IBM

IBM World Trade

Z10-0003-0

SALES and SYSTEMS GUIDE

# Handbook for the Design of Small Communications-Based Systems

This Handbook provides assistance to the field in selling, designing, and installing small communications-based systems (CBS). A communications-based system is defined as basically a data processing system which has some communications capability and which uses a simple communications application program.

The book is divided into seven sections. Section 1 defines the scope of the Handbook. Section 2 deals with the characteristics of a small communications-based system. General systems design aspects are dealt with in Section 3. Detailed systems design, based on the IBM 2740 and 1050 terminals, 2260 display terminals, and STR terminals in specific applications, are illustrated in Sections 4 to 6 inclusive. A logical approach to preliminary implementation planning is given in Section 7. It is assumed that individual countries will supply, as a source of reference material, an appendix that is country-oriented and relates to communications tariffs, configurations, and other pertinent material.

Copies of this and other IBM publications can be obtained through IBM branch offices.
Address comments regarding the contents of this publication to:
IBM World Trade Corporation
Product Marketing Department
821 United Nations Plaza
New York, New York 10017

December, 1966

IBM CONFIDENTIAL

CONTENTS

This book is a design guide for small communications-based systems (CBS). Such a system is defined as basically a data processing system which has some communications capability and which uses a simple communications application program.

The Handbook is intended for IBM salesmen and systems engineers in World Trade countries who have had little or no experience in communications-based systems.

The objectives of the Handbook are:

1. To assist IBM salesmen and systems engineers in selling, designing, and installing data processing systems with some communications capability, by providing appropriate educational material and factual information.

2. To reduce the effort in time and manpower involved in selling, designing, and installing such systems by providing simplified techniques and procedures, including graphs, formulas, and examples.

3. It is intended as an educational tool for IBM personnel and, indirectly, for prospects and customers. While the book is for IBM Internal Use Only, relevant material from it can be used for customer courses, lectures, or seminars, provided the salesman or systems engineer adjusts his presentation of the material to suit the audience.

The book is divided into seven sections. Section 1 defines the scope of the Handbook. Since IBM has such a wide range of computers, terminals and programming systems for Tele-processing applications, the scope of a design handbook such as this, which can deal with only a part of this vast range, must be precisely defined. This section also contains information concerning the use of the Handbook.

Section 2 deals with the various characteristics of a small communications-based system, the hardware used, the communications environment, programming philosophy, and the method of control involved.

General systems design aspects are dealt with in Section 3, leading logically on to the detailed communications-based system design of the IBM 2740 and 1050 terminals, 2260 display terminals, and high-speed synchronous transmit and receive terminals (STR) in specific applications in Sections 4 to 6, inclusive.

A logical approach to preliminary implementation planning is given in Section 7.

Appendix A is a source of reference material that is country-oriented and relates to communications tariffs, configurations, and other material which may need to be used on numerous occasions.

Sections 1 and 2 and the first three subsections of Section 3 are concerned with general information about communications-based systems and are intended for salesmen and systems engineers. The remainder of Section 3 and all the other sections are intended primarily for systems engineers, but may also be useful to salesmen interested in the details of hardware, programming, system design, and implementation planning.

The Handbook does not replace other IBM literature, but combines information from a variety of documents in order to make the subjects clearer and simpler to use. When necessary, reference is made to publications which should be used in conjunction with a particular subsection.

The Handbook does not cover basic information on System/360 concepts, facilities or instruction sets, nor any programming or descriptions relating to input/output devices other than those used in communications-based systems. The reader should have some prior knowledge of basic System/360 programming concepts and hardware.

The value of the Handbook will lie in the reader's appreciation of the gaps which it fills in the range of available manuals, the intelligent use which he makes of it, and the time which it will save him in his marketing and design procedures.

## SCOPE OF THE HANDBOOK IN THE CONTEXT OF A SMALL CBS

In order to decide what type of system would be considered within the scope of the Handbook and to define a small CBS, a survey was carried out on the communications-based systems installed and on order in Europe as of May, 1966. It was found that over 90% of the systems contained IBM 1050, 2740, 2260, and STR terminals, and that most of the computers involved were System/360 Models 30, 40, or 50. The majority of the systems contained fewer than ten terminals. The indications are that this trend will continue for some time to come.

## DEFINITION OF A SMALL COMMUNICATIONS-BASED SYSTEM FOR USE IN THIS HANDBOOK

To clarify further the small communications-based system definition, the following limitations are placed on such a system:

1. The base CPU is a Model 30, 40 or 50.

2. The system is simplex and cannot be part of a full or partially duplexed configuration. There are no duplicate central units for reliability.

3. The time specified for transmitting the last character of an input message from a terminal to the receiving of the first character of the associated output message at the receiving terminal is greater than or equal to two seconds.

4. The application does not include remote stacked-job processing.

5. The system must use standard Type I programming support for the IBM 1050, 2740, and 2260, without modification.

   (a) IBM 2740, 1050, and the remote 2260 must use BTAM with DOS or BTAM with SPS as its communications control program.*

   (b) The local IBM 2260 must use BTAM with DOS, or GPS with SPS as its communications control program.

6. The system must use standard Type I and Type II programming support for the synchronous transmit/receive terminals, which are STRAM with DOS, or STRAM with OS. There may not be any modification of the IBM programs.

7. The only non-IBM T/P equipment in the configuration is modems and/or communication lines.

8. The system must not require T/P RSDP's.

9. The system must use only announced products.

10. No concentrators or line-switching are used.

11. There are no stringent reliability requirements. Unexpected system interruptions and down-times should not cause serious difficulties for the customer's operation.

12. No concurrent shared data set or shared file use is required.

13. The communications portion of any of the configurations must be similar to those displayed in the following chart.

| SYSTEM | AVERAGE NUMBER OF LINES | AVERAGE NUMBER OF TERMINALS | ADDED I/O FOR COMMUNICATIONS |
|---|---|---|---|
| 1050/ 2740 | 6 | 10 | 2311 OR 2314 DISK FILES AND SUPERVISOR TERMINAL |
| 2260 | 2 | 16 X 2260<br>2 X 2848 | 2311 OR 2314 DISK FILES AND SUPERVISOR TERMINAL |
| STR | 2 | 2 | AUXILIARY STORAGE |

This chart is offered as a guide and indicates to the user the type of system that may be categorized as small. Certain design sections of the Handbook develop the detailed limitations of a small communications-based system, showing the user how to calculate the required system parameters so that he may determine whether his system falls within the bounds of a small CBS.

*SPS is Option 2 (MFT) of the Operating System.

SECTION 2.

CHARACTERISTICS OF A COMMUNICATIONS-BASED SYSTEM

## 2.1 ADVANTAGES OF A CBS

Before proceeding to the functional, hardware and programming characteristics, it is appropriate to consider some marketing aspects of a communications-based system and their advantages to customers.

Any prospect or customer who is considering a communications-based system will undoubtedly raise the questions: "What advantages does it have?" "How will it assist me in my business?"

Let us first consider why these methods have been introduced. Data transmission terminals connected to communication lines reduce the time delays involved in transferring data from remote locations to a computer and in returning the resultant processed output to those distant locations. In some cases there can be a radical reduction in the time involved from a matter of days to mere seconds.

By reducing these time delays, communications-based systems have three main advantages for the user, any one of which is sufficient justification for the use of the system:

- Increased efficiency
- Improved service
- Reduced costs

If the application has been well considered and the system design and installation work effectively carried out, a communications-based system should supply all three.

We turn now to how we can make the prospective customer aware of the advantages which such systems will give him over his existing operation.

### JUSTIFYING A COMMUNICATIONS-BASED SYSTEM

### A. Improved Service

For companies which provide a service to their customers, a reduction of time delays can result in vastly improved service by significantly decreasing waiting time.

In other applications, customer inquiries at remote locations could be handled on the spot; customer reservations confirmed in seconds; or customer credit ratings acknowledged within minutes.

### B. Centralized Operations and Control

Suppose a customer has a number of decentralized batch-processing centers in several remote locations where data is prepared for input to a central system. With the addition of Teleprocessing methods to the system, the customer can control his remote locations from his data processing headquarters. All his remote locations would conform to a common operational procedure in preparing input to the system and following a specific method of operation after the various outputs were received. Duplication of operations would be reduced.

### C. Accurate Control and Current Updating of Files

A communications-based system allows files to be updated almost as soon as an issue, receipt, or other similar event takes place. With accurate files, a customer can quickly learn the up-to-the minute status of his business and take timely action on it.

In a credit system, for example, each individual's account can be updated as transactions occur. In a sales order entry system, where goods are stocked in a number of decentralized warehouses, the CBS would reduce inventory costs to a minimum, with maximum availability of goods.

### D. Management Reports

Managers can readily obtain up-to-date reports based on current data, at specified terminals or central high-speed printers. They can be quickly notified about abnormal conditions, or critical events occurring in their business.

### E. Reduced Clerical Effort

This advantage is also a part of centralized operations and control. The communications-based system eliminates the need for human operations between the original recording functions and the ultimate data processing.

### F. Elimination of Costly Errors

Errors caused by transcribing the same misinformation from form to form are eliminated.

A communications-based system performs its own checking of all transmitted information to ensure that it has not been distorted during transmission. Records transmitted are checked to ensure that no information is lost.

G. Rapid Two-Way Communication

Through a CBS, the customer is able to converse with the CPU from remote locations and get immediate response. He can send broadcast messages from the center to the terminals, notifying them simultaneously of changes in company policy, prices, etc.

These are only a few of the benefits which a communications-based system can bring to a customer's business. But each of these relates specifically to one of the three chief advantages:

- Increased efficiency
- Better service
- Reduced costs

## MAJOR ADVANTAGES OF AN IBM COMMUNICATIONS-BASED SYSTEM

What are the major advantages of an IBM CBS? This question can be answered by reference to the points made below. It is pointed out here that this subsection is only a summary of the advantages and that the user of the Handbook must become familiar with the hardware and programming in order to be able to discuss the specific advantages.

### A. IBM Hardware Geared to Communications

The IBM T/P product line provides for all aspects and functions of a communications-based system. The hardware also covers a wide range of communications-based system types, many of which are designed to be industry-oriented. IBM can also build custom-made terminals to meet individual customer requirements. IBM modems can interface World Trade communications facilities.

### B. IBM Programming Packages, Types I and II, for T/P

IBM has major programming support for communications-based systems. There are a number of Type I and II programs for the System/360, such as:

| | |
|---|---|
| BTAM/DOS | Type I |
| BTAM/OS | Type I |
| QTAM/OS | Type I |
| QTAM/DOS | Type I |
| STRAM/DOS | Type II |
| STRAM/OS | Type II |
| CCAP | (Communications Control Application Program) Type II |
| PARS | (Programmed Airline Reservation System) Type II |

In most cases, all these packages enable the customer to control his network, analyze the input transaction leader, log and queue transactions, and prepare and send the output.

### C. IBM Online Diagnostics

Some of our existing programming systems include extensive online diagnostics (otherwise known as error recovery and detection programs). BTAM and QTAM will have these in the near future, while CCAP and other Type II programs already have them. These facilities are described in detail in subsection 3. 1 of the Handbook.

Online diagnostics provide checkpoint and restart, error statistics, error recovery, line test, terminal test, isolation of errors, and contribute to operator awareness.

### D. IBM Hardware Detection of Normal and Abnormal Operations

The IBM CBS's -- in particular, the transmission control units -- are capable of detecting normal and abnormal operating conditions, such as:

- Detection of a negative, positive, or no-answer to a poll or selection of a terminal;
- Detection of an open line;
- Detection of errors caused between the TCU and the data set;
- Echo-checking to ensure that a proper bit was sent down the communications line from the TCU;
- Stop bit error detection: recognizes that, after receiving a start bit and the required number of data bits for a character, a stop bit has not been received;
- Automatically builds the LRC character and performs the compare, which results in an interrupt if there is no compare;
- Automatically detects parity errors within characters received from the terminal;
- Automatically recognizes End-of-Block and End-of-Transmission characters.

This is not an exhaustive list, but it is given here to emphasize advantages to the customer. These are useful sales points, especially if the customer is technically knowledgable in the communications area.

E. Expandability

The customer should be informed that a CBS is designed for expandability. IBM's T/P products are modular and therefore can be added to a basic CBS and increase its capability. The system can start out initially as a simple inquiry system and be expanded to a full-blown communications-based system. The ease of expandability of a CBS can accommodate the increase in traffic, customer locations, and/or online applications.

CBS's are usually planned and installed in phases to parallel customer growth. Thus, when a vendor closes an order for the first phase of a total CBS, he usually acquires most of the orders for all other phases of the CBS.

IBM can supply a total system, from the terminal and data set all the way to the multiplexors and central processors. The customer can be shown how IBM equipment can meet his system's specifications and grow with his needs without major changes to his initial CBS configuration. A small, simple inquiry system can be the beginning of a large, complex CBS.

CONCLUSION

By reading the Handbook in entirety, the salesman and systems engineer will undoubtedly think up additional advantages and sales points of a CBS and also better understand those which have already been described.

## 2. 2 FUNCTIONAL CHARACTERISTICS

In this chapter we will survey briefly a typical CBS in order to identify the functions of each component, while keeping the broad picture in mind at all times.

Figure 1 is a block diagram of a typical, simple CBS. The major components are described below.

### 1. TERMINAL

The terminal is the remote input/output unit, usually located where the source information to the system is created. The terminals are usually sending as well as receiving devices; or, in other words, can handle both input and output. Input can be in the form of keyboard entry, paper tape, cards and magnetic tape, displays and paper tape.

The function of the terminal is to convert the input into electrical pulses or signals to be placed on the communications channel, directly or via a data set. It also interprets incoming signals and control information, selects the proper output unit, and places the signals in an intelligible form, such as a printed page, a punched card, a displayed image, etc.

### 2. MODEM* (or Data Set)

In most cases, a data set is required. The data set's function is to provide compatibility between I/O equipment and communication facilities. The data set is a modulation/demodulation device. In other words, the data set converts** electrical pulses from a device into a form suitable for a communications facility (see Subsection 2. 2. 3).

### 3. COMMUNICATION CHANNEL

The communication channel provides the transmission medium between the central and remote locations.

### 4. MULTIPLEXOR AND DATA ADAPTERS

At the receiving end, the electrical signals representing information must enter the CPU for further handling.

In a normal CBS, the traffic is originated at remote locations, each independently of the other, as dictated by actual local activities. The operator must generally wait a short time before he can use the terminal. This may require that the system collect messages on a real-time basis: messages enter the CPU simultaneously on a random basis -- as many as there are active lines. Simultaneous reading of all lines is achieved through multiplexing of the data through the TCU and the multiplexor channel. (Refer to Subsection 3. 6. )

From the standpoint of programming, constant attention must be given to the lines in order to:

(1) service each character flowing on an

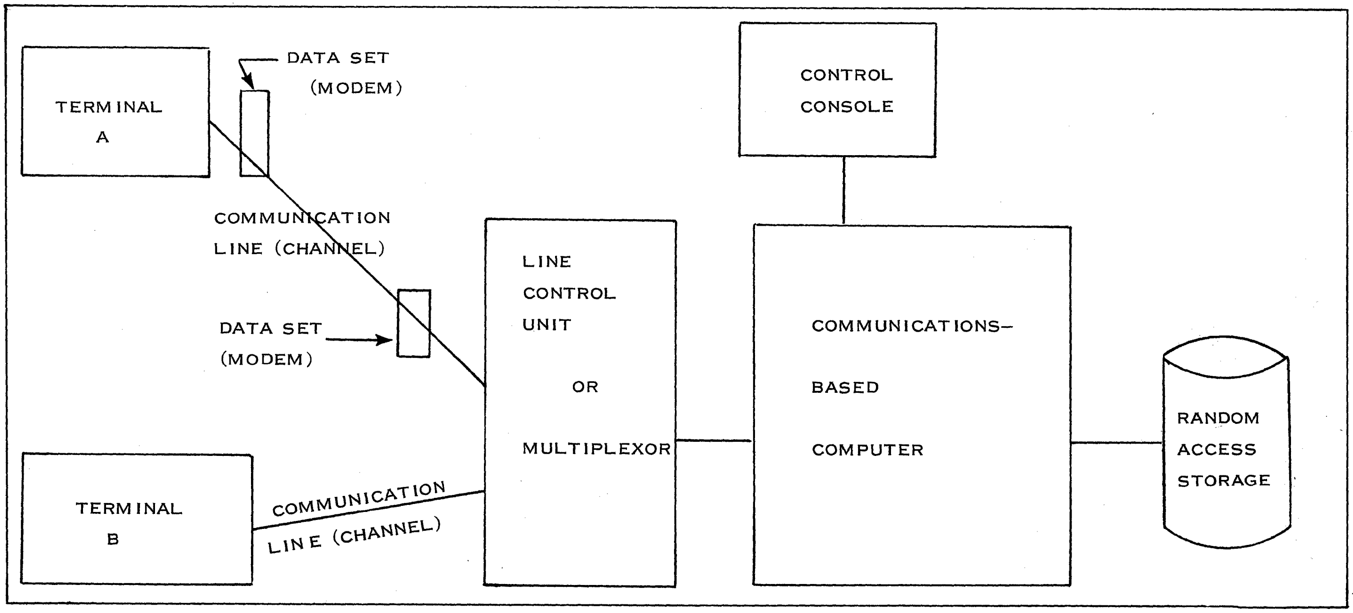

TERMINAL A
DATA SET (MODEM)
COMMUNICATION LINE (CHANNEL)
DATA SET (MODEM)
TERMINAL B
COMMUNICATION LINE (CHANNEL)
LINE CONTROL UNIT OR MULTIPLEXOR
CONTROL CONSOLE
COMMUNICATIONS-BASED COMPUTER
RANDOM ACCESS STORAGE

Figure 1. Communications-Based System or Online System

* Do not confuse data set with a "set of data" used in the terminology of OS/360.

** Can generally be called "signal converter": data set or modem or line adapter for telephone lines; "level converter" for telegraph lines.

active line;

(2) maintain terminal polling;

(3) maintain complete awareness of any abnormal condition on the communications network.

5. CPU

As a result of the above traffic flow, the CPU must assemble each message, identify it for protection purposes, queue it for further analysis and processing, and possibly log it for future retrieval or statistical purposes. The functions of line, terminal control and message control are to define the special job to be done by both the front-end hardware and the programming.

6. CONTROL CONSOLE (Supervisor Terminal)

The control console performs the communications functions between the operator at the center and the system (CPU), e. g.:

Receiving messages from the computer:

(1) to keep the operator (or supervisor) informed of manual operations and status of the system [traffic, statistics, etc.]~~)~~;

(2) to alert him of abnormal conditions, failures, errors, etc.

Sending, by operator control, corrective information in the form of formatted messages to the computer.

7. DIRECT ACCESS STORAGE DEVICES (DASD)

These can be a disk, drum, or large-capacity core storage. At present, disk is the usual direct access storage device used in a communications-based system. The disk is used for:

a. Line or terminal queues
b. Logging
c. Checkpoints
d. Error recovery routines
e. Data files for application programs
f. Control program

To illustrate the relationship of all the components in a CBS to each other, we will now trace a message through the system (see Figure 1).

A message is originated at Terminal A and has as its destination Terminal B. The message can be prepared offline on paper tape and placed in the paper tape reader of a terminal.

The message is then transmitted over the communication channel through modems, if required, to the multiplexor or the data adapter unit. This unit, in turn, recognizes certain control characters and interrupts the communications-based computer when End-of-Message is detected. The computer now performs a header analysis and makes sure the message is valid. The computer recognizes the destination of the message. The message is then placed in a history log and also in a queue on disk for the line that services Terminal B.

The computer monitors the status of the Terminal B line and the status of Terminal B itself. When the line is free and the terminal ready to receive, the computer reads the message for Terminal B from disk into core. The proper commands are sent to the multiplexor (or data adapter unit) and the message is sent over the communication channel through a modem, if necessary, to Terminal B. The terminal accepts the message and prints it out on its typewriter.

CONCLUSION

So far, we have only mentioned the components which constitute a CBS. In the next chapter we present functional details concerning them. We will cover, first, the media which tie the remote and local components together: telecommunications concepts and facilities.

## 2. 3. TELECOMMUNICATIONS ENVIRONMENT

As part of a communications system, the communication network plays a significant role, in which the technical characteristics of the lines are an important consideration. Transmission techniques are quite sophisticated and require maximum understanding for proper design of a CBS.

It is not intended here to explain in complete detail the theory of transmission techniques and media, and the purpose of this chapter is:

(1) To assist in designing a CBS;

(2) To provide enough information to facilitate communication with telecommunications specialists acting either for customers or PTT organizations.

### 2. 3. 1 COMMUNICATIONS FACILITIES AND CHARACTERISTICS

A communications-based system consists of a number of input/output devices, usually in geographically dispersed locations, connected by one or more communication lines.

Our purpose is to define the term "communication line" and inform the reader of the problems inherent in communication lines. All types of communication lines are included under the concept of "communication facilities".

Communication facilities offered by common carriers (PTT's or operating agencies) are examined first as to their technical aspects and then with regard to their availability and performance.

#### Technical Facts About Transmission Lines

The primary purpose of telecommunication lines is to transmit speech and telegraphic signals. Existing transmission lines are designed and classified according to the criteria of "line quality". "Quality" refers to speech quality and is not always completely in accordance with the criteria for data transmission. These criteria are as follows:

1. Bandwidth

A simple example may serve as a definition:

A radio transmitter is usually located many miles away from a studio and the sound (speech or music) is transmitted to the transmitter through telephone lines. The sound seems quite good when we hear it on the radio. Now, let us suppose that during a program (a play, for instance) one actor has to answer the telephone. The sound we hear from the telephone is poor. The difference arises from the different qualities of the lines used for telephone and radio. The normal telephone line and the radio telephone line have different bandwidths.

A complex sound is composed of many different elementary frequencies; and if they are not all transmitted, the received sound is strongly affected, or distorted. The decrease in magnitude of a signal in transmission between points is called "attenuation".

Expressed technically, if we draw a curve with the frequencies in the X axis and the ratio of the energy received over the energy transmitted (attenuation) in the Y axis, we obtain a "bandwidth" curve of the shape shown in Figure 2.

Attenuation is measured in decibels (db), whose mathematical expression is:

$$n \text{ db} = -20 \log \frac{V_2 \text{ (volts)}}{V_1 \text{ (volts)}}$$

For instance, 3 db is the attenuation of energy by a factor of 2. If instead of log 10 we use log e

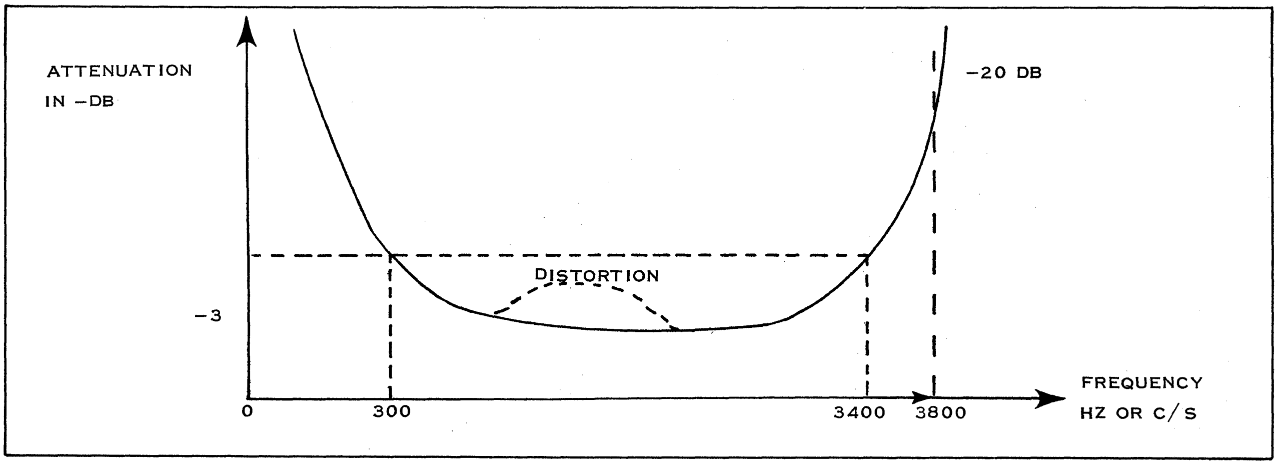


Figure 2. Bandwidth Curve of a Voice-Grade Line

the same formula is expressed in decinepers*:

1 db = 1. 15 decineper

If the reference signal is constant and equal to 1 mwatt in an impedance of 600 ohms, the db is referenced as a dbm.

The example in Figure 2 is the bandwidth of a voice-grade line.

Amplitude distortion (also called frequency distortion) arises from the uneven attenuation of the different frequencies within the bandwidth. Attenuation is greater for higher frequencies than lower, and the difference in attenuation is directly proportional to the length of the transmission line.

2. Envelope Delay

This is the curve of the propagation time for the different frequencies within the bandwidth. If the propagation time differs too much for some frequencies, it can result in a distortion of the phase of the signal transmitted (phase distortion). This kind of distortion is very critical for high-speed data transmission. The delay interval will increase in direct proportion to the length of the circuit.

3. Miscellaneous

Other causes of distortion or improper transmission are echo, extraneous noises, cross-talk, and radio interference. Other signals on the same line can interfere either directly or by induction from adjacent wires and so disturb the transmission.

## Improvement of Transmission Lines

All the above distortions may be suppressed or at least decreased by PTT technical services. The existing means of improving transmission are briefly summarized as follows:

Attenuation can be improved by using wide band amplifiers and loaded cables. (Inductances are inserted on a loaded cable at fixed intervals.) This improves the range of the cable, but decreases its bandwidth.

Amplitude distortion can be suppressed by the insertion of selective amplifiers which reinforce the attenuated frequencies.

Phase distortion can be improved by using devices called phase equalizers. These devices slow down the non-delayed frequencies in order to delay the whole band equally.

Cross-talk can be decreased by balancing the adjacent pairs in the same cable.

Extraneous noises can be attenuated by devices called filters which permit the attenuation of either low frequencies (high-pass filters) or high frequencies (low-pass filters), or permit the transmission of only one range of frequencies (band-pass filters).

## Line Service and Facilities

According to the criteria for data transmission as described in the previous paragraphs, transmission lines can be classified with respect to the maximum speed at which the data can be transmitted without a high percentage of errors. Speed is expressed in bits per second. **

Common carriers (PTT's and/or operating agencies) provide facilities for communication services:

Either switched service for comparatively light users, billed on a message basis --

Example: Public telephone for voice communication
Public telex for telegraph communication

or

Leased service, where facilities are leased by a customer for private use between several of his locations, where traffic is sufficiently heavy to warrant such an installation.

Economically, there is a break-even point between the two services, for example:

A telegraphic leased line, for a speed of up to 50 bps, between Stuttgart and Paris, two distant locations, would cost $1,400 per month, or the equivalent amount of 2,000 units of 3-minute telex calls.

Similarly, a "voice-grade" line or telephone line between Stuttgart and Paris would cost $2,948 per month, or the equivalent amount of 4/3 times 2,000 units of 3-minute telephone calls.

There are other special services, on a part-time or full-time basis, such as a "program" service for the transmission of broadcasting communications, television network, etc.

Since requirements for data transmission are somewhat different from those of normal facilities, data services are being introduced by the PTT's both on a switched and leased basis.

* The neper is a unit frequently used by PTT's.

** "Baud" means bits/second within the context of this book.

Speed

Speeds now available or which could be made available in the near future vary from:

50 bps on a telex network;<br>
50, 75 and 100 bps on leased telegraphic lines;<br>
200 bps either on the public switched telephone network or a special telegraphic switched network (refer to possibility of Datex in Germany) and on a leased special telegraphic network;<br>
600/1200 bps on the public switched telephone network;<br>
1200, 1800, 2000, 2400, and possibly 4800 bps on leased voice-grade lines (with special conditioning in the case of the higher speeds, called "equalization") and high speeds, such as 40. 8 kilobits/second (40,800 bits/ second) on special facilities called "primary groups" (a group of 12 voice-grade lines); and 200 kilobits/second on other special "groups".

From the viewpoint of the design of a total system, considerations of interest to a customer would be tariff structures of different services and the trade-off available between service and cost, including the selection of terminals and the actual speeds of transmission.

Line Operation

Waarom altijd > 2 draden

Depending on how a single line is constituted, many operations are possible:

1. Two Wires - One Channel: Non-reversible (Figure 3)

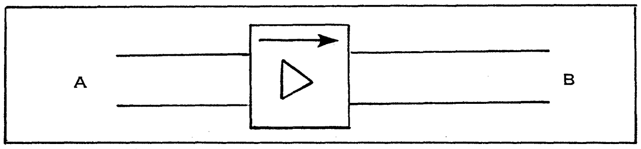


Figure 3. Single Line: Two Wires, One Channel: Non-reversible

Transmission (also called "simplex transmission") is possible only from A to B, because a non-reversible device, such as an amplifier, is inserted on the line.

2. Two Wires - Two Channels (can also be n channels): Non-reversible (Figure 4)

The representation is the same as above, but the bandwidth can be divided in order to provide two or more different paths.

3. Two Wires - One Channel: Reversible (Figure 5)

Transmission (also called "half-duplex transmission") is possible either from A to B or from B to A, but not in both directions simultaneously.

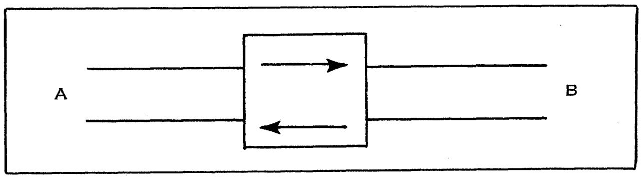


Figure 5. Single Line: Two Wires, One Channel: Reversible

4. Two Wires - Two Channels (can also be n channels) (Figure 4)

This is the same as 2 above. The bandwidth, being divided into n channels, permits independent operation of the channels. With this type of circuit, simultaneous transmission in both directions is also possible (called "full-duplex transmission"). In example 2, the two channels are made to operate in the same direction, whereas in this case they operate in opposite directions.

5. Four Wires (Figure 6)

This line is composed of two simplex lines and permits full-duplex operation.

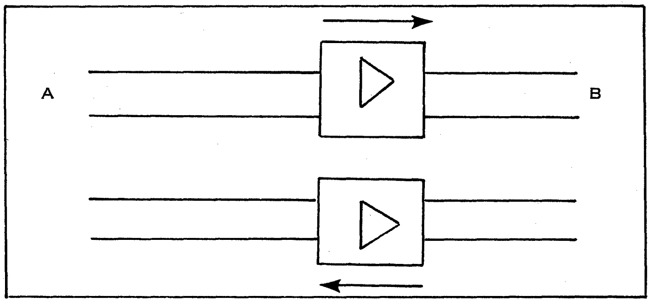


Figure 6. Single Line: Four Wires

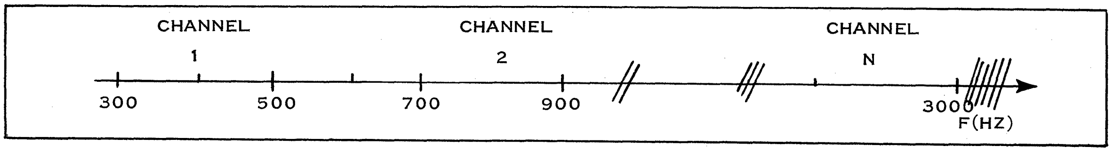


Figure 4. Single Line: Two Wires, Two Channels

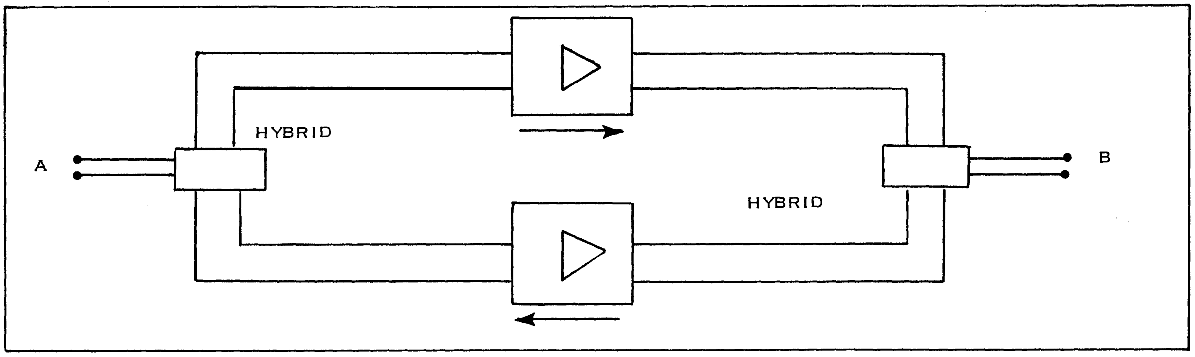


Figure 7. Long-Distance Line

Line Termination (Figure 7)

Long-distance lines are always composed of four-wire circuits, but the terminations may be either two-wire or four-wire ("optional"). The transformation is made with a special device called a hybrid transformer.

Problems relating to two- or four-wire circuits deal with overall stability and attenuation of the circuits.

It is important to note that transmission lines provide the possibility of half- or full-duplex, but this operation is determined only by the terminal/ modem connected to the line.

### 2. 3. 2 SIGNAL CHARACTERISTICS

Digital data is represented by a succession of pulses. Data terminal equipment cannot ordinarily be directly connected to a telephone network, since direct current signals delivered by data terminal equipment cannot be transmitted over it. Therefore, it is necessary to convert these signals into a form suitable for transmission. This conversion, which is called modulation, is made by a modulator. Conversely, at the receiving station, signals coming from the line must be demodulated by a demodulator before they are transmitted to the receiving data terminal equipment.

Modulation is the process by which certain characteristics of a wave are modified in accordance with a characteristic of another wave or signal.

Demodulation is the reverse process: the original signal is reconstructed from a modulated wave.

There are several types of modulation (see Figure 8):

Amplitude modulation: the modified characteristic is the amplitude of a carrier signal.

Frequency modulation: two separate frequencies are used to represent binary information. The lower frequency represents bit 1; the higher frequency represents bit 0. This is also called FSK (Frequency Shift Keying).

Phase modulation: the phase of the carrier is reversed with one pass from bit 0 to bit 1.

Other types of modulation, such as Vestigial Side Band (VSB) and Digital Modulation, use more advanced techniques, but all these techniques have the same purpose: to concentrate most of the energy spectrum within the bandwidth and provide protection against noise and distortion.

### 2. 3. 3 MODEMS

We have shown the need for an intermediate device between the digital signal and the communication line. This device is called a modem (contraction of modulator-demodulator).

A modem accepts the digital signal, converts it into an appropriate analog signal for transmission over the line. Conversely, it accepts the analog signal from the line and reconverts it into a digital signal (demodulation), and presents it to the terminal through an interface. The modem is also called a data set and when integrated and housed in the machine itself, is called a line adapter.

A modem consists, functionally, of:

- a transmitter
- a receiver
- control circuitry

The transmitter modulates the incoming data signals, using a carrier frequency generated in the modem itself.

The receiver reconstructs the signal, amplifies it when necessary, extracts from it the original data and presents it to the terminal equipment through the interface. Some non-IBM modems also include a device which checks and corrects the

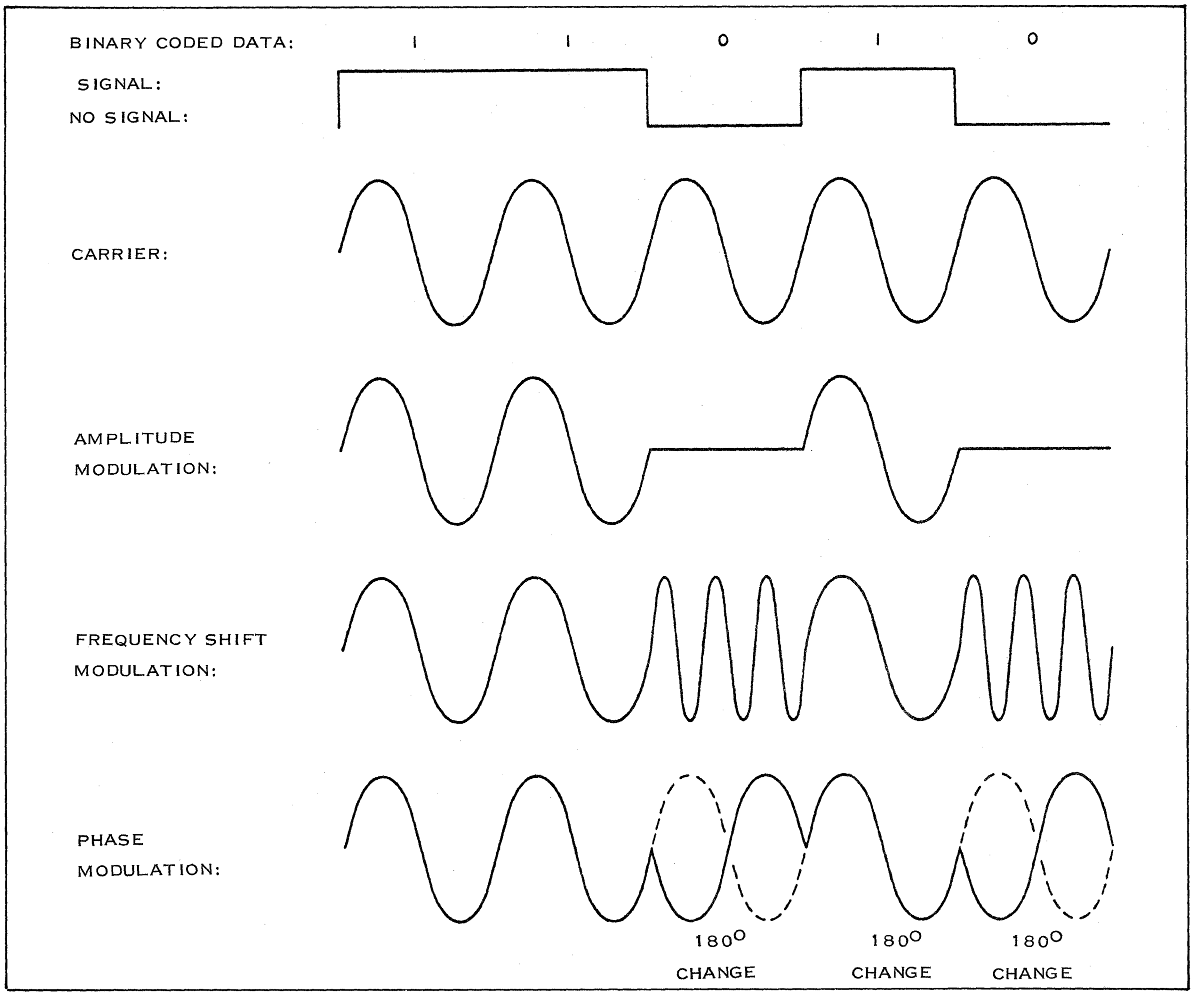


Figure 8. Modulation Techniques

received data. IBM modems are transparent (passive) and transmission checking is done by the terminal itself.

The control circuitry performs various functions, such as permitting conversation with the terminal equipment, checking its status, controlling the connection to the line when a switched network is involved, etc.

## Line Interface

As we have seen in the paragraphs on Line Facilities, the line connected to the modem may have 2 or 4 wires. Operation of the modem depends on how the line is constituted.

In the IBM 3977 modem, operating with only one carrier, the transmitting part and the receiving part can operate either alternately in half-duplex mode (according to the terminal equipment capability) or a 4-wire line can be connected directly to the transmitter and receiver (respectively) and permit full-duplex operation. The hybrid is not used.

In the IBM 3976 modem, however, only the two-wire connection is standard, but provides two sub-channels (2 different frequency bands, 2 carriers) within the voice-grade bandwidth. This modem permits either full- or half-duplex operation, depending on terminal equipment capability. A 4-wire connection is provided on this modem on an RSDP basis, but only for special applications. This is required by the PTT, for instance for multi-drop configurations.

## Terminal Interface

25 interface circuits are defined for connecting a modem to a terminal. They are listed in the document issued under recommendation V24 SP A100E of the C. C. I. T. T.* This committeė is one of the I. T. U. (International Telecommunication Union) groups representing all countries concerned with standardization of telecommunications.

*Comité Consultatif International Télégraphique et Téléphonique - International Telephone and Telegraph Consulting Committee.

Availability of Modems

For the availability of modems, see Appendix A.

### 2. 3. 4 HOMOLOGATION AND PTT RELATIONS

To connect communications equipment to a communications line, other than a customer-owned, in-house communications facility, it must first be approved by the common carrier (generally the PTT). The approval procedure is called "homologation".

A device is homologated after the PTT has satisfied itself that it will interface and interwork properly with the communications facilities, will not cause power surges, and complies with all other local regulations. Whether the modem alone or all additional equipment connected to it is subject to homologation depends on the regulations of the country concerned.

### 2. 3. 5 TRANSMISSION ASPECTS

Transmission Mode

Data characters can be transmitted over communication lines in two ways:

- serial-by-character, parallel-by-bit,

  or
- serial-by-character, serial-by-bit

In parallel mode, each constituent bit of a character is sent simultaneously over the line. The bit-carrying medium can be a multi-wire line (IBM 357) or a set of fixed frequencies used on the same line (IBM 1001).

In serial mode, the character is coded as mentioned above and sent on to the line, bit by bit. This operation is achieved by a special circuit called a "serializer-deserializer".

Terminal Mode of Operation

As we have seen, there are different types of terminals which can be classified according to mode of operation: simplex, half-duplex, and full-duplex.

There are two kinds of full-duplex operation, depending on the terminal operation:

- Data characters may be transmitted simultaneously in both directions;

- Data characters are sent in one direction and only control characters are sent in the other direction. Only this kind of full-duplex operation is used in IBM terminals (refer to Modem Operation, 2. 3. 3., and 2-4 Wire Circuits, 2. 3. 1).

### 2. 3. 6 TRANSMISSION TECHNIQUES AND SYNCHRONIZATION

In order for the receiving hardware to properly detect each bit and character, some way of synchronizing the receiving and transmitting ends is needed. Two synchronization schemes can generally be used:

- "Start-stop" for low speed, when each character is framed by a start and a stop bit. This can be used under certain conditions up to a speed of 1200 bps.

- "Synchronous" for high speeds, where clocking information is either transmitted with the data or derived from the data for proper detection.

In the start-stop mode, before and between characters, the line is in "1" condition (Mark). Transmission of a character is started by the transition from "1" to "0" condition (start bit). The receiving terminal detects this start bit and starts its clock (oscillator), which stops when the stop bit is received. Here, the bit sampling is synchronized on stop-start transition.

This method is used for low- and medium-speed transmission, because it permits the use of a cheaper clock (the oscillator does not need very high accuracy). It applies to the IBM 1050 at 200 bps; the 2260 at 1200 bps, etc.

Synchronous transmission techniques are described fully in Section 6. It should, however, be pointed out that the transmitter and receiver are synchronized in a constant manner and that sampling of the incoming bits can vary in order to correct line distortions or oscillator drift.

### 2. 3. 7 CODING

Coding is defined as a system of rules and conventions according to which the data signal forming a block is formed, transmitted, and processed.

The code is formed from the letter of the alphabet which represents the character itself, plus one or more bits used for checking the character (in that case, the code is said to be "redundant").

A complete character flowing across a communication channel may also include additional bits used for transmission synchronization purposes (start and stop bits). Those characters are not included in the code itself.

The most frequently used codes are the following:

Baudot Code

Each character is represented by 5 bits. For the purpose of transmission synchronization, each 5-bit character is preceded by a start bit (0 or Space) and followed by a stop bit (1 or Mark). This code is still used in telegraphy and also in a telex network, but it is not often used for data transmission because it provides no checking facilities. In fact, there are no invalid characters in the Baudot Code when alphameric characters are used. (Refer to Section 2. 3. 8.)

Binary Coded Decimal Code (BCD)

Each character is composed of six bits (BA8421). One additional bit (C) is added to check the parity of each character and one or two stop bits are added to the character to ensure synchronization of the terminals. The IBM 1050 System uses this as its transmission code.

ASCII (American Standard Code for Information Interchange)
or
ISCII (International Standard Code for Information Interchange)

The ASCII code structure has seven bits plus the parity bit, or eight bits plus the parity bit (ASCII-8). The latter is very similar to the internal System/360 code. The IBM 2260, for example, uses the ASCII-8 code in remote applications.

4-Out-Of-8 Code

The 4-Out-Of-8 character code structure has a total of 8 bits, made up of four 0 bits and four 1 bits. This code is used in such IBM STR devices as the IBM 1009, 1013, 1978, 2020, etc.

Other Codes

Other codes can be used for data transmission, with different schemes for automatic error detection and even automatic error correction.

### 2. 3. 8 ERROR CHECKING AND MESSAGE PROTECTION

The basic concept in error checking is the desired accuracy of transmission compared with quality (error statistics) provided by the channel itself. If an error affects only one bit, a simple parity check will be sufficient to detect an error; if a great many errors occur, a more elaborate detection method must be used.

IBM data transmission techniques provide only error detection. Correction is automatically achieved by retransmitting erroneous messages. Depending on transmission speed, IBM uses the following error-detection techniques:

1. One single-bit error: The parity of each character is checked. For example, in addition to the six bits of the BCD code, a C bit is used for this purpose. This check is called "vertical redundancy check" (VRC). (For the 4/8 code, an internal check is provided by the code itself, since one character must always include four 0 bits and four 1 bits.)

2. Double-bit errors: The parity of a whole block of characters is checked. This check character is called "longitudinal redundancy check" (LRC). It is sent after the message and compared with the one first received at the receiving station.

Other techniques may be used to detect and correct errors:

- Cyclic Codes

- Double transmission: The message is sent twice and the two messages are compared. The message may also be sent back to the transmitter for comparison (loop operation). The disadvantage of this method is that transmission time is greatly increased.

Cyclic codes and double transmission are not used by IBM at present.

For further information concerning this chapter, refer to Data Communications Concepts and Communications Facilities, Form E20-8158.

## 2. 4. TERMINAL AND LINE CONTROL

Information sent from remote terminals to a central processor, and vice versa, via communication channels is called a "message" or a "transaction". This unit of information will be used later as the basic quantum of activity of a communications-based system. Input, or received messages, is defined as coming from the terminals to the central processor, and output (or transmitted) messages are sent from the processor to the terminals.

Since these messages or transactions are the basic input or output data to and from the CPU and carry the required information, it is essential that the system be assured of complete control of this process to avoid errors or losses. Perfect control by the system (supervision and operator) requires:

- complete identification of the message and its origin;
- awareness of the message status while transiting in the system;
- complete knowledge of terminal and line status for availability and maintenance purposes.

### 2. 4. 1 MESSAGE CONTROL

For the CPU to perform message control, the message carries information relative to its processing. The message consists of a header (which contains such information) and the text of the information to be transmitted.

The message is received as a whole, but if it is too long, it can be segmented. The segment is determined by such considerations as buffer size and other hardware or operating characteristics.

Other special characters are included in the message and permit control of the line by the CPU.

### 2. 4. 2 TERMINAL AND LINE CONTROL

This control has the following functions:

- controlling the use of the line facility by terminals;
- controlling the procedure of communications between terminals and either a master station or a CPU.

The terminals can be in one of the two following modes:

- control mode for preparation of the data link before actual data transmission;
- text mode, during which significant data is sent.

In order to communicate with the terminals, the CPU (or master station) uses either polling or addressing.

- Polling is the operation by which the CPU calls the terminal and allows it to transmit.
- Addressing is the operation by which the CPU asks the terminal to get ready to receive information from it.

For further details, refer to the Systems sections, 4. 1 and 5. 4 (1050 and 2260), where this oneration is fully described.

## 2. 5 DATA TRANSMISSION UNITS

The IBM 2701, 2702, and 2703 Transmission Control Units permit the connection of a CPU to communication lines.

Depending on the composition of these units, the following maximum number of lines can be attached:

| | | |
|---|---|---|
| For the | 2701 | 4 lines |
| | 2702 | 31 lines |
| | 2703 | 176 lines |

Since the 2703 cannot be considered part of a small CBS, only the 2701 and 2702 will be discussed.

### 2. 5. 1 2701 DATA ADAPTER: FUNCTIONAL DESCRIPTION (Refer to Subsection 3. 6. 2 for a detailed description)

The 2701 is composed essentially of a channel interface (CHIF); transmission interface converter (XIC); and transmission adapter (XA).

The channel interface is common to all lines; each line requires a coupled (XIC-XA). (For limitations on the maximum number permissible, based on types of XA's, see the Configurator, Subsection 3. 6. 4.)

The 2701 has the following functions:

- Interfaces to MPX or selector channel;
- Interfaces to transmission lines;
- Directs and controls information flow between the channel and the terminals;
- Operates the conversion bit/byte (serialization/deserialization);
- Permits attachment of various terminals, such as start-stop, STR, and 2260's;
- Recognizes special characters;
- Performs special checks: VRC, LRC;
- Detects abnormal conditions: time out;
- Permits hardware checking as far as the interface with the data set (diagnostic register and wraparound command);
- Can operate up to 40. 8 kb in STR mode.

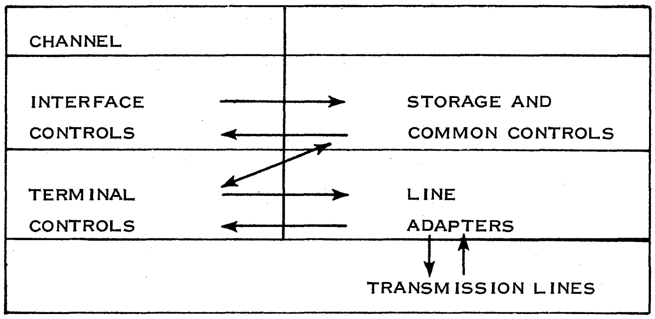


Figure 10. 2702 Terminal Controls

### 2. 5. 2 2702 TRANSMISSION CONTROL UNIT: FUNCTIONAL DESCRIPTION (Refer to Subsection 3. 6. 3 for detailed description)

The IBM 2702 is composed essentially of the following sections: interface control; storage and common control; terminal controls; line adapters.

The maximum permissible number of lines attached to the 2702 is 31. (For limitations and arrangement of the different terminal controls, see Configurators, subsection 3. 6. 4.)

The 2702 has the following functions:

- Interfaces to the multiplexor channel;
- Interfaces to transmission lines, but has more capabilities than the 2701 regarding the line adapters that can be connected;
- Directs and controls information flow between the MPX channel and the terminals;
- Operates the bit/byte conversion (serialization/deserialization);
- Permits attachment of various terminals but has some limitations as compared with the 2701: the Type III terminal (2260 and STR) cannot be attached to a 2702 and the maximum speed allowed on the 2702 is 600 bps;
- Recognizes special characters;
- Performs special checkings: VRC, LRC;
  Detects abnormal conditions: time out;
  Permits hardware checking as far as the interface with the data set.

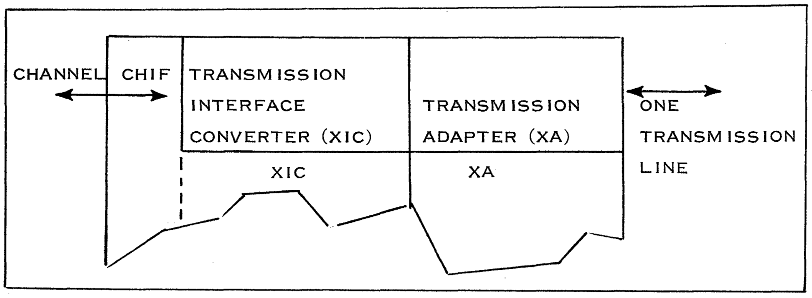

CHANNEL
CHIF
TRANSMISSION INTERFACE CONVERTER (XIC)
TRANSMISSION ADAPTER (XA)
ONE TRANSMISSION LINE
XIC
XA

Figure 9. 2701 Functional Design

## 2.6 PROGRAMMING SUPPORT

A computer system is composed of hardware (CPU, channels, control and I/O units) and programming (programmed instructions for jobs that the system has to perform). Programming, again, can be divided into application programs and programming support. Whereas application programs are based wholly on specific jobs, programming support performs such standard functions as:

| | |
|---|---|
| • Job scheduling | Control Programs |
| • Operator communication | |
| • Data accessing | |
| • I/O and task control | |
| • Language translation | Processors (Compilers) |
| • Program generation | |
| • Sorting and merging | Utility Programs |
| • Transferring data from one I/O to another | |

Figure 11 is a diagram showing programming support.

In this subsection, the control program area of the programming support is examined as far as it is concerned with small communications-based systems.

### 2. 6. 1 DISK OPERATING SYSTEM (DOS) VERSION II

DOS II replaces DOS I and provides some additional optional functions. These additional functions are mainly multi-programming (the control of three independent partitions) and telecommunications (the interface to the Basic Telecommunications Access Method [BTAM]). The use of these functions requires a 32K machine with the Storage Protection Feature.

The three partitions:

Foreground 1, Foreground 2, and
Background

are completely independent. They cannot interface with each other, as they are protected by different storage protection keys. In effect, each partition functions as though it is a separate computer sharing the arithmetic and logical capabilities of the CPU with programs operating in the other partitions. There is no means of direct communication between partitions. However, the access to a file is possible from more than one partition. In the case of simultaneous updating of a file by two programs, exclusive control has to be programmed by the user.

Priority of execution of the different programs is:

1. Supervisor
2. Attention routine (operator commands)
3. Foreground 1
4. Foreground 2
5. Background

As soon as one program enters the wait state or is completed, control is given to the one with the next lower priority. The use of multi-programming and telecommunication functions requires the following restrictions:

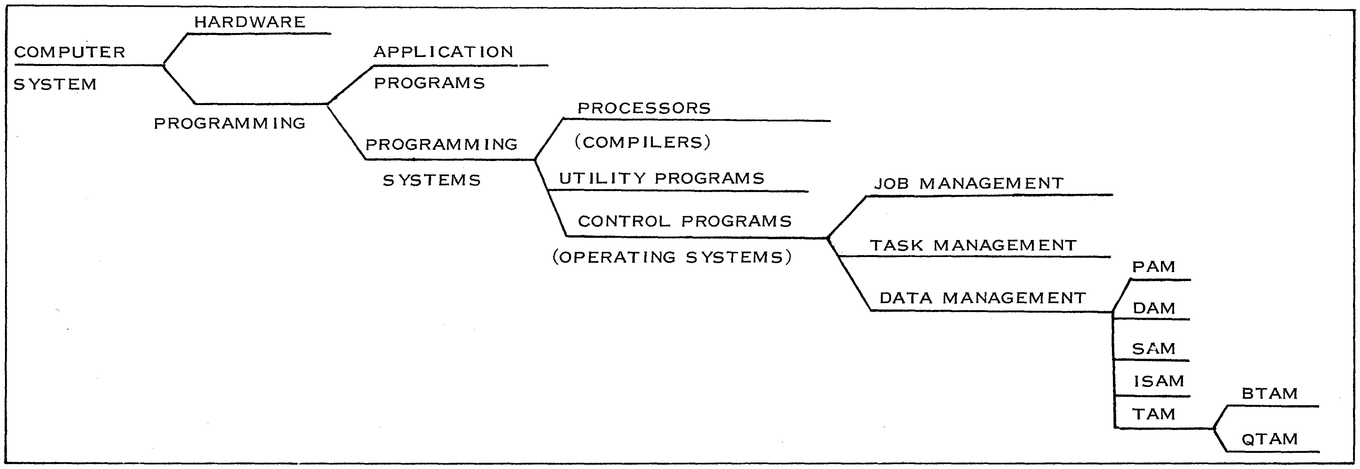


Figure 11. Programming Support

1. FORTRAN and PL 1 programs as well as IBM processing programs operate only as background programs.

2. There must be no burst mode devices on the multiplexor channel.

3. The 2701 is supported only on the MPX channel.

4. The timer can be assigned to only one partition for time-interval use (time-of-day may be used in all partitions).

### Use of Partitions

| | |
|---|---|
| Background - no restrictions | Compiler (same levels as in DOS)<br>Application programs in PL1, FORTRAN, COBOL and Assembler,<br>Utilities, and<br>T/P programs using BTAM |
| Foreground 1 and 2 | Application programs in Assembler language,<br>Special DOS utilities generated by macros, and<br>T/P programs (in Assembler) using BTAM (COBOL for processing is possible if necessary linkages are provided) |

### Size of Partitions

At generation time, the required number (1 - 3) of partitions of the necessary size is generated. The operator has the choice of altering the size or number of partitions by issuing a special job control command through the console.

8K is used for the supervisor (the final version with error recovery procedures may possibly use 10K).

10K is the minimum size of the background partition, as this is the size of the job scheduler.

Multiples of 2K can be given to the foreground partitions. All partitions, including the supervisor, must be within a multiple of 2K limits because this is the smallest area for a storage protection key.

### Initiating of Programs

In the background partition, the job scheduler initiates the jobs sequentially from the job input queue.

In the foreground partitions, the jobs are initiated and terminated by the operator through the console, using special job control commands.

## 2. 6. 2 BASIC TELECOMMUNICATIONS ACCESS METHOD (BTAM) IN DOS

Like the other data access methods of System/360, BTAM is designed to ease reading and writing of data, in this case from and to terminals.

This is done by providing the necessary macros for all such functions as read, write, addressing, and polling.

As in most other cases, a terminal determines when a message has to be read in. The polling technique or something similar is used. The result of a read command will, therefore, not always be reading of data, but may sometimes be a negative response of a terminal. It is an additional function of the telecommunications access method to handle these negative responses to polling. In order to poll the lines during processing of messages, BTAM provides an internal supervisor as part of the assembled program. This supervisor has control over all T/P interrupts. A negative response to polling results in initiation of polling of the next terminal in the polling list. An EOB or EOT interrupt indicating completion of a line read or write operation during message-processing results only in posting of the interruption until the message-processing is terminated. The reason is that no overlapped message-processing is possible in a single program (task) environment, as it is provided by the Disk Operating System.

### Supported Terminals

- IBM 1030, 1050, 1060, 2740, AT&T 83B3, and WU 115A through a 2701, 2702, or 2703 on non-switched lines (BTAM can support all these terminals at the same time, as long as each line has only one kind of terminal.)
- IBM 1050, 2740, and TWX 33/35 on a switched line through a 2701, 2702, or 2703
- IBM 7770/7772 attached to the multiplexor channel (BTAM does not support audio response units in the same system with other terminals)
- IBM 2260 remote terminal through a 2701 on non-switched lines
- IBM 2260 local terminal attached to the multiplexor or selector channel through a 2848 Control Unit.

## BTAM Functions

The main functions which BTAM provides are:

- Transmitting and receiving messages
- Polling or addressing terminals according to the user's established lists, and changing the polling or addressing scheme at execution time
- Directing buffer pools to send or receive variable-length messages
- Allowing programs with lower priority to use the CPU during wait times
- Providing online terminal test facilities
- Checking for transmission errors, recovering from temporary errors, posting permanent error conditions, and writing error messages
- Automatic answering or dialing to establish a connection to a remote terminal

## User's Responsibilities

The responsibilities of the user are:

- To define his system
- To write the message-processing programs, including:
  - Line control, using BTAM macros
  - Code translation
  - Time and date stamping, if required
  - Header analysis
  - Message queuing and logging
  - Any additional error procedures required

## Definition of the System

For a BTAM coding sequence, this example shows where and in what sequence the definition statements are used. All operands and the program itself are omitted to simplify the example:

```
         START
BEGIN    BALR
         USING
         •
         •
         User's program using BTAM
         •
         •
         READ      MF = L
         DTFBT
         DTFBTND  } may be assembled
         DFTRMLST } separately as a set
         BTMOD    - should be assembled
                    separately by itself
         END        BEGIN
```

A Data Event Control Block (DECB) is generated per line by the READ macro with the entry MF = L.

A Define the File Table (DTF Table) is generated per line group by the DTFBT macro.

After the last DTFBT macro, a DTFBTND macro must follow.

Polling and addressing lists are generated by the DFTRMLST macro.

BTAM logic (internal supervisor) is generated by the BTMOD macro.

## BTAM Macros

A. Declarative Macros*

| | |
|---|---|
| BTMOD | generates logic module of BTAM |
| DTFBT | generates DFT Tables |
| DTFBTND | defines the end of DTF's |
| DFTRMLST | generates polling and addressing lists |
| CHGNTRY | changes polling and addressing lists |
| STATB | generates statistical tables |

B. Transient Macros

| | |
|---|---|
| OPEN | activates lines |
| CLOSE | deactivates lines |

C. Line Activity Macros

| | | |
|---|---|---|
| READ | | polls and reads data from a line |
| WRITE | | addresses and writes data to a line |
| CONTROL | | enables or disables a line (for auto-answering) |
| RESETPL | 1. | stops polling for a leased line using a wraparound list |
| | 2. | drops the connection of a |

*The READ/WRITE/CONTROL macros are also used for generation of DECB's.

| | |
|---|---|
| | switched line, gives control to the program with next lower priority |
| TWAIT | Test-wait for testing purposes |

D. Buffer Management Macros

| | |
|---|---|
| REQBUF | Requests a buffer from the buffer pool |
| RELBUF | Releases a buffer to the buffer pool |

Polling and Addressing Lists

There are two kinds of lists in BTAM:

- Wraparound lists
- Open lists

Open lists are used only for addressing.

For polling, both can be used. After polling the last terminal in a wraparound list, the first entry is used as the next polling address. In the open list, polling stops at the end of the list.

Buffering

To reduce core requirements for I/O areas in large systems and for variable message length, BTAM provides a means for creating buffer pools per line or a common buffer pool.

BTAM automatically chains the buffer segments, but queuing of messages must be programmed by the user. Returning the buffers after processing is also the user's responsibility.

Error Recovery Procedures

Error recovery procedures are provided as an option on a line-group basis for all BTAM-supported devices.

The recovery procedure depends upon the type of error encountered. The types of errors are:

- Temporary errors: those which are recovered as a result of retrying the erroneous operation;
- Permanent errors: those which resist a predetermined number of retries to recover from the erroneous operation, or those which are considered to be unrecoverable.

If the procedure used is unsuccessful or if no attempt is made to recover from the error, a specific error code is posted in the error field of the DECB and a system-to-operator message is printed on the SYSLOG device.

Error Statistics

Error statistics are offered on a line-group basis for most of the BTAM-supported devices, when error recovery procedures are performed.

A threshold is defined per line. When the number of retries reaches the threshold, a system-to-operator message is printed on the 1052 console typewriter.

Example: If threshold is coded: THRESH = (3, 4, 100) in the STATB macro instruction, a system-to-operator message must be printed when in each hundred transmissions three data checks or four other errors have occurred.

Note: No error statistics are kept for the IBM 2260 local, 7770/7772, or the 2740 without station control.

### 2. 6. 3 QUEUED TELECOMMUNICATIONS ACCESS METHOD (QTAM) IN DOS

QTAM is a generalized input/output control system that extends the techniques of Logical IOCS to the telecommunications environment. Files accessed by the problem programmer are queues of messages coming in from, or going out to, remote terminals via communication lines. Even though the time and order of arrival and departure of messages to and from the CPU are unpredictable, the programmer can handle them as if they were organized sequentially.

Unlike other commonly used access methods, QTAM furnishes far more than the mechanics for input/output operations. In addition to the standard GET/PUT macro instruction support for message-processing programs, QTAM provides an extremely high-level and flexible message-control language. QTAM-supplied macro instructions can be used to construct a complete message-control program that controls the flow of message traffic from one remote terminal to another (message-switching application), and between remote terminals and any message-processing programs (message-processing applications). Therefore, an installation-oriented message-control program can be written in a relatively short period of time instead of the many months previously required for such a programming task.

A QTAM message-control program is generated from a number of Assembler macro instructions coded by the programmer. Although the Assembler macro generator is used, the process followed is similar to that used by a high-level compiler. A QTAM message-control program is open-ended: that is, the user can include functions not provided through the QTAM language by employing DOS control program macro instructions.

A generated message-control program is necessarily completely device-dependent, with all communication lines and terminals identified to the system. Through file-definition and control-information macro instructions, the user specifies his hardware configuration and the main storage areas (buffers) required for his application. These macros generate the table and lists of control information that define the environment of the system for the QTAM logic. The number and size of the buffers required are specified by the user, and are one of the primary resources in the telecommunications system. The buffers are allocated to a common buffer pool from which QTAM automatically and dynamically uses them in accordance with immediate requirements.

QTAM logic modules are also provided for many procedural functions, such as message-code translating, routing of messages, and error checking. By the selection of appropriate macro instructions, the user specifies which of these QTAM logic modules are to be incorporated into his message-control program. In this way, the system can be tailored to the exact requirements of the application being supported.

The message-processing program services of QTAM enable a programmer to process messages from a telecommunications network with the same easy-to-use macro instructions that he uses for his local input/output devices. When a QTAM message-control program is used to perform the input/output operations, a device-independent message-processing program can be written. The programmer is, in effect, completely freed from the time and device-dependent aspects of the telecommunications environment. By using some other access method for a sequentially organized file, the user can completely write and test his message-processing programs without ever running them in the telecommunications environment. Then, by simply changing the definition of a DTF table, he can reassemble to operate under QTAM control.

In order for a QTAM message-control program to handle the flow of message data between a message-processing program and the remote terminals in a system, it is necessary that there be an interface between the message-control program and the message-processing program. QTAM provides this in the form of queues in the message-control task and macros (GET and PUT) in the message-processing program, which enable the programmer to access these queues.

### 2. 6. 4 OPERATING SYSTEM OPTION 2 (OS - SPS)

Option 2 provides up to four partitions in main storage, in which the corresponding number of separately scheduled jobs resides concurrently, each in its own fixed partition. When processing is temporarily halted in one partition, as for example, during a wait for I/O completion, processing is switched to the next-lower-priority partition to take advantage of the temporary delay.

Partitions are initially defined at system generation. The highest priority cannot be assigned to a task without being associated with the partition of highest priority. A task's priority is determined solely by the partition in which it operates. At nucleus initialization (NIP) time, the number of partitions may be redefined up to the limit established at system generation. For example, four partitions may be specified at system generation; at NIP time, the system may be redefined as containing one, two, or three partitions. Furthermore, the size of each partition can be redefined at NIP.

The first job of the job input stream is initiated in the highest-priority partition. It signals that operations may begin in the next-lower-priority partition by issuing a WAITR macro instruction. This causes the next job in the input stream to be scheduled into the next-lower-priority partition. (The system uses a single job input stream as the source of jobs for all partitions.) The new job, similarly, issues a WAITR to activate the next partition: this process continues until all partitions are active.

With two or more jobs residing in main storage at the same time, normal task-switching occurs. Processing proceeds in the highest-priority ready partition until that partition waits; control is then given to the next-higher-priority ready partition until that partition also waits, in which case control passes to a lower-priority partition, or until the wait in the higher-priority partition is satisfied, causing that partition to regain control.

The system is designed for use with single "unending" job steps in all higher-priority partitions, with a batch job in the partition of lowest priority. The precedence of job types that should be observed is shown in Figure 12, and begins with the highest priority.

Option 2 provides from two to five task-control blocks, in priority order as follows: Master Scheduler (the standard system TCB), Partition 0 (highest-priority partition), and so on, to Partition 3 (lowest-priority partition). Associated with each boundary box, except that for the Master Scheduler, is a partition-related scheduler control block that

| PROGRAM | PARTITION | PRIORITY | K BYTES |
|---|---|---|---|
| SPV | - | 0 | 20 - 25 |
| TP | 0 | 1 | ≥18 * |
| GRAPH | 1 | 2 | ≥18 |
| SPOOL | 2 | 3 | ≥18 |
| BATCH | 3 | 4 | ≥18 |

Figure 12. Priority of Job Types

*If the 44K job schedule is used, all partitions must be greater than 44K bytes.

includes an Event Control Block (ECB). The status of the ECB determines whether the scheduler can operate in the associated partition. The partition-related scheduler control block also provides a "save" area in which a partition's job queue pointers are recorded before the scheduler is assigned to a lower-priority partition.

## 2. 6. 5 BTAM IN SPS

### Use of BTAM in SPS

BTAM may be used in any (one or more) partition. Normally, a T/P program using BTAM should run in the highest-priority partition. In case there are some STR lines or graphics with a higher priority to be handled, the program using BTAM can run just as well in a lower-priority partition. Apart from a degradation in performance, there are no known restrictions on operating in this mode. However, it would not be advisable to try to support "uncontrolled" terminals such as the 2740 without the Station Control Feature in other than the high-priority partition.

Operator communication (for example, finding out which lines or terminals are working or how long the message queue is) should be made through a local terminal so as not to delay the T/P task by using the console, which has a higher priority. Therefore, in BTAM only a local terminal can be used for these functions.

### Supported Terminals

BTAM in SPS supports the same terminals as in DOS, except the 7770/7772.

### BTAM Functions and User's Responsibilities

See DOS - BTAM (page 21).

### Definition of the System

The DCB macro is used in OS to define the lines, terminals, buffers, etc. The other declarative macros, such as DFTRMLST or READ with MF = L, are the same as in DOS-BTAM.

### Buffering

SPS-BTAM uses PCI interrupt to dynamically chain buffers as data are being received (DOS-BTAM chains the buffers together before the read starts).

## 2. 6. 6 QTAM IN SPS

QTAM under Option 2 requires two partitions, one for the message-control task and one for the message-processing task. These must be the two highest-priority jobs in the system and are loaded in the normal manner by the job scheduler.

Option 2 provides inter-partition communication for QTAM only.

QTAM requires this ability in order to communicate the message-control task with the message-processing task. QTAM does this without violating protection. Communication is via a pointer in the Communications Vector Table.

For QTAM-provided functions, refer to subsection 2. 6. 3, where QTAM in DOS is described.

## SECTION 3.

## GENERAL DESIGN TECHNIQUES

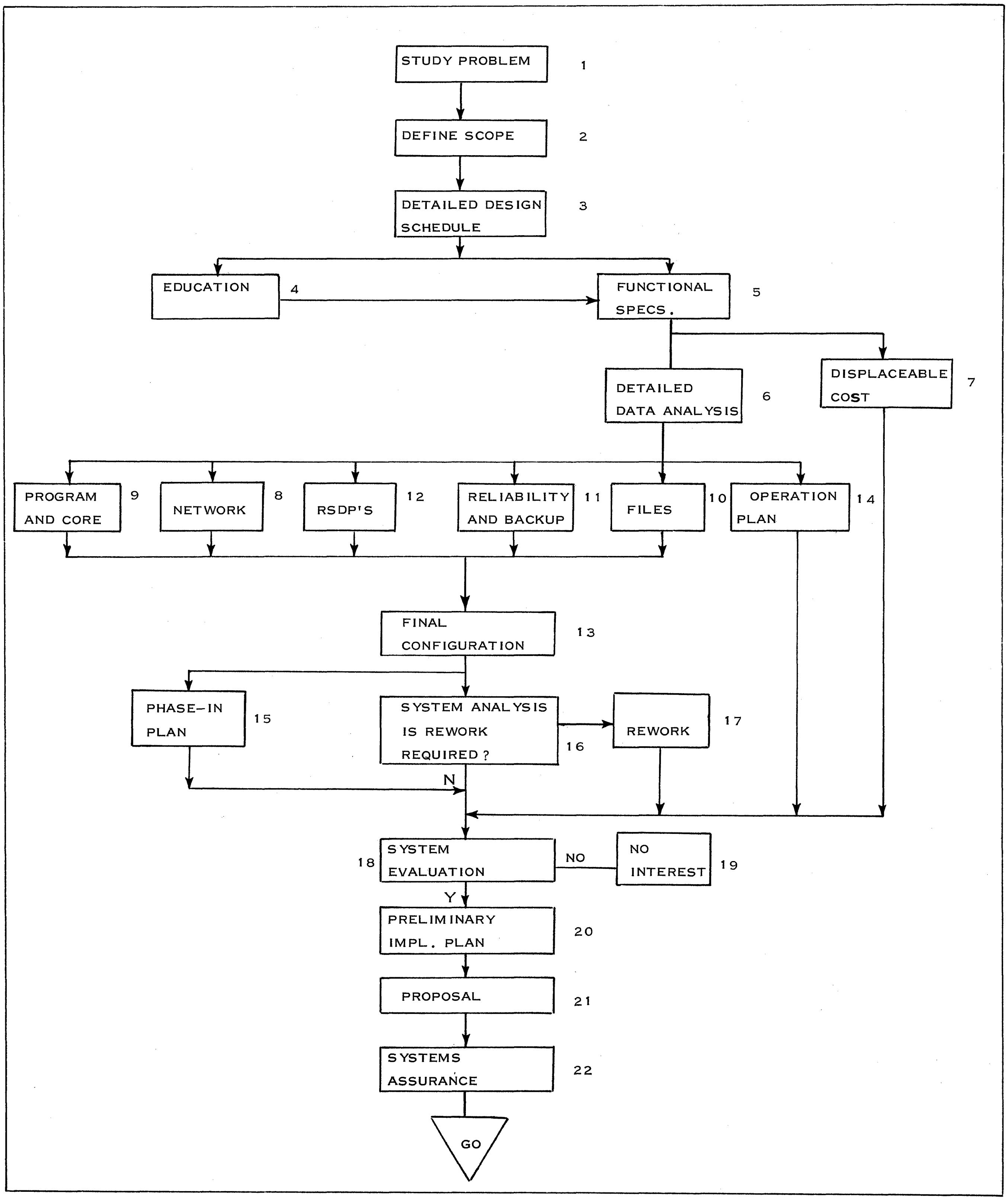


Figure 13. Summary Overall Design Chart

## 3. 1 GENERAL SYSTEM DESIGN APPROACH

This subsection of the Handbook covers the important areas in the design of a communications-based system, and a method of approach to the design of a CBS.

The system design can be summarized in three statements:

1. It is a detailed study of the customer's problem to create a basis for functional specifications.

2. It is the collection of data in sufficient detail to perform a proper traffic analysis.

3. It is the complete system design, which results in a proposal to the customer.

The purpose of this subsection is, further, to show the user the various tasks that are necessary in order to achieve proper system design. It is not intended to go into great detail, but only to describe each task briefly and to give the reasons why some tasks are dependent on others.

This subsection also discusses how the major design techniques (described in detail in subsections 3. 2 to 3. 9) relate to each other.

### 3. 1. 1 DESCRIPTION AND INTERRELATION OF THE SYSTEM DESIGN TASKS

This part of the "General System Design Approach" covers the steps involved in the system design, including a description of all the major design tasks and their interrelations, along with an example of a schedule and manpower requirements.

The relationship of the major tasks in the system design is shown in Figure 13. This chart does not include every task, as shown in the checklist in 3. 1 . 2, but illustrates the major tasks into which all other tasks in the checklist can be categorized.

A summary description of the tasks shown in Figure 13 is as follows:

1. Study Customer's Problem

The purpose of studying the customer's job is to develop sufficient explicit information to permit an actual design for the system.

2. Defining the Scope of the Job

This involves deciding what major functions should be included in the design considerations. With proper understanding of the customer's problem, and knowledge of System/360 hardware and programming, the designer is able to choose the functions to be included or considered as a basis for the functional objectives and system design.

3. Detailed Design Schedule

The purpose of the schedule is to define the scope of the job task, since it defines the major functions that must be included in the design considerations. Figure 14 is an example of a schedule for a relatively small CBS, similar to the 1050 example described in subsection 4. 5. In this case, it took two men six weeks to complete the system design and proposal-writing for a CBS based on the information presented in this Handbook.

Before presenting the example, the following assumptions must be made, based on Figure 14:

- Coordination is assumed to be exercised by A.

- Systems Assurance time has to be added to the schedule (see Systems Assurance, subsection 3. 12).

- The schedule does not consider travel time, responsibilities for other projects, etc. The schedule also assumes previous experience or education of the personnel involved in a CBS design.

The schedule shown in Figure 14 is merely an example. It is up to the system design team leader to estimate his own time and manpower requirements in light of the scope of the application, experience, and education of the design team, and which design tasks are necessary to complete the particular system design.

Scheduling also includes the assignment of tasks and the documentation plan. To make the user aware of the need for documentation, the following reasons are stated:

1. To have all the information required by Systems Assurance readily available and arranged in proper order for review.

2. Because a CBS proposal usually contains technical information and information required by the customer specification, the detailed documentation of the system design is a necessity.

| TASK | MEN | EDUCATION | WEEKS |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|
|  |  |  | 1 | 2 | 3 | 4 | 5 | 6 |
| EDUCATION | ALL |  |  |  |  |  |  |  |
| FUNCTIONAL SPECIFICATIONS | ALL |  | ■ |  |  |  |  |  |
| DATA COLLECTION | A |  |  | ■ |  |  |  |  |
| DATA REDUCTION | A |  |  | ■ |  |  |  |  |
| NETWORK DESIGN | A |  |  | ■ |  |  |  |  |
| TERMINAL DESIGN | A | AS NEEDED (2 DAYS TO 3 WEEKS) |  |  | ■ |  |  |  |
| FILE ORGANIZATION | A |  |  |  | ■ | ■ |  |  |
| SYSTEM ANALYSIS | A |  |  |  |  | ■ |  |  |
| PROGRAM DESIGN | B |  |  | ■ | ■ |  |  |  |
| PROGRAM ESTIMATES | B |  |  |  |  | ■ |  |  |
| CONFIGURATION | B |  |  |  |  | ■ |  |  |
| PHASE-IN PLAN | B |  |  |  |  | ■ |  |  |
| OPER. PLAN AND IMPL. PLAN | A |  |  |  |  |  | ■ |  |
| DISPL. COST AND SYS. EVALUATION | B |  |  |  |  |  | ■ |  |
| PREPARE PROPOSAL | ALL |  |  |  |  |  |  | ■ |

*TIME AND MANPOWER REQUIREMENTS FOR SYSTEM DESIGN

|  | NO. OF MEN | TIME |
|---|---|---|
| MINIMUM | 2 | 3 WEEKS |
| MAXIMUM | 3 | 6 WEEKS |

* These are only estimates and should be used for planning purposes.

Figure 14. Example of a System Design for a Small Communications-Based System: Schedule

Naturally, to create a documentation plan, documentation specifications are required to indicate the method of documentation, and how the documentation is filed.

4. Education and Training

For a small CBS that comes within the scope of the Handbook, the team must first be trained and become familiar with this Handbook before proceeding to designing the system. The qualifications of the design team must include familiarity with the System/360, including configurations and programming packages). The design team must also be knowledgable in communications-based system design. This covers the area of communications concepts, terminals, network design, communications programming, and system analysis. Finally, the design team personnel must be familiar with the customer's application and the system specifications. These qualifications are based on experience and education.

The subjects that must be studied are listed below:

(1) System/360, including the 2701 and 2702

(2) System/360 Operating Systems, including BTAM

(3) System/360 direct access devices

(4) Communications and network design

(5) Terminals

(6) Programming for a CBS using System/360

(7) CBS design techniques and approaches

Knowledge of these subjects can be acquired from manuals, guides, and condensed system design courses. The Handbook has been developed to consolidate all the topics mentioned in the preceding list and also to reduce time spent on education and training of the design team.

5. Functional Specifications (Refer to Subsection 3. 2)

The functional specifications are the basis for the flowcharts of the problem program and the design of the total system. It is preferable that the education of the design group be completed prior to the functional specifications, but if this is not possible, the team can be trained on the job, at least to some extent.

Functional specifications can be developed either by the customer alone, or jointly with IBM personnel.

6. Data Collection and Analysis (Refer to Subsection 3. 2)

This includes data collection, data reduction, and traffic analysis. This task has direct input from the functional specifications and the definition of the scope of the job. It also includes the collection and analysis of statistical data in terms of number of records per file and number of terminal locations.

There are various methods of collecting dynamic data, but generally some type of data-collection form is used. Data is collected and analyzed, based on the following parameters:

(a) Message Types

Traffic statistics must be taken for each message type in the system.

(b) Message Length

The average maximum and minimum length of each message type must be included as part of the datataking. Besides the number of characters of text, the designer must add the number of control characters required for each message type.

(c) Messages Per Time Period

When collecting data, time periods must be fixed in terms of minutes, hours, days, and even months, depending on what the application requires.

(d) Input/Output Ratio

Messages should be segregated according to whether they are sent from the terminal to the system, or from the system to the terminal. Input/output ratio means the ratio of input messages to output messages. In most cases, this ratio is one-to-one or one to a number greater than one.

(e) Priorities

The number of priorities required for the particular application must be known and the data collected must be placed in its individual priority category.

(f) Peak Traffic

Data collection results in pinpointing the peak traffic loads during specific time periods. This, in most cases, is what the system is designed to handle.

(g) Data Reduction

Once the data has been taken, it must be reduced to intelligible form. The data is reduced to number of types of messages in and out per terminal, and number and types of messages in and out of the central system. To be meaningful, all data must be reduced to a common base.

(h) Growth Projections

The data that is taken is usually current. The system that will be designed should, of course, provide for the future. A realistic growth factor must be applied to the data in order that projections can be made for the future traffic load of the system.

(i) Response Time Requirements

This is usually included in the specifications, but the designer must make sure that for each message type there are response requirements, whether they be seconds, milliseconds, minutes, or days. (This topic is expanded in subsection 3: 2. )

7. Displaceable Cost

The displaceable cost is made up from the cost elements of the customer's existing system that may be eliminated with the installation of the proposed system. The techniques for determining these costs for communications-based systems are the same as those used for traditional batch-type systems.

The functional specifications define in what areas we can make a displaceable cost study.

8. Final Network Design (Refer to Subsection 3. 5)

Before this task can be accomplished, the traffic analysis must be completed. This task consists of the final terminal design and the selection of the proper communications facility. The objective of network design is to provide the means for getting data from remote points into the central with minimum cost and with an acceptable standard of performance.

(a) Terminal Selection

All possible terminal types must be considered in a CBS design. Before choosing a proper terminal design, constraints such as printing quality and input and output media must be considered.

With the field of possible terminals narrowed, the next step is to consider the load-carrying capabilities of the various possible choices; or, in other words, how much traffic a terminal can realistically handle based on its speed and reliability characteristics.

The final choice of a terminal will be made only after the cost of the communications facility attached to the terminal is considered, as the total cost of the remote complex is both a function of the terminal and the facility cost.

(b) Communications Facilities (Refer to Subection 2. 3)

Prior to completing a network design, a communications facility must be selected, or at least the number of communications facilities should be reduced to two or three. Communications facilities differ in the following way:

> Speed; whether they are leased or public switched networks; whether they are supplied by one or more common carriers; whether they are privately owned; half- or full-duplex, multi-point or point-to-point; type of line control, and modems required.

9. Program Design and Core Estimates (Refer to Subsections 2. 6, 3. 7, and 3. 11)

This task is definitely based on the functional specifications task and input from the data-taking and traffic analysis tasks. It must also be considered in conjunction with the final network design and file design.

All the tasks mentioned in the preceding paragraph are input to the program design and must also be considered in conjunction with it. These tasks define tables, programs, the philosophy of program design, buffer requirements, program handling of buffer, and line and terminal control programs.

10. Logical File Organization (Refer to Subsection 3. 8)

This task is dependent upon the data collection and traffic analysis, which helps to define the size and type of file. Program design and file organization are directly dependent upon each other and one cannot be completed without the other if a good system design is to be attained. In this task, the formats of the records are established in detail and the actual number and types of physical files selected.

The principal parameters affecting the file selection are:

- Volume of information to be stored

- Response time to retrieve or store information

- cost of storing and accessing this information.

Other parameters affecting the selection of the file that are applications-oriented are:

- Applicability of available file organizational and maintenance programming packages
- Nature of the activity against the file
- Compatibility of data rates with CPU
- Necessity or desirability of write-checking some or all information written into the file
- Growth potential
- Reliability requirements
- Characteristics of information to be stored, such as pure numeric information
- Extent of hardware utilization

11. Reliability: Duplexing, Back-Up Plan, and Restart and Recovery (Refer to Subsections 3. 3 and 3. 10)

A manual back-up plan and restart and recovery programs may be necessary for a small CBS. The functional specifications are the basis for their development. The back-up plan and restart and recovery procedures must be developed with the program and core estimate task, inasmuch as additional programs may be necessary for restart and recovery.

In general, a small CBS is a simplex system and therefore does not have stringent reliability requirements. If the customer is interested in reliability, the reliability requirements must be established at functional specification time.

To make a reliability analysis, the aid of Systems Assurance is required. Reliability parameters for IBM equipment are not available to the field, but are available to Systems Assurance. This obviously makes it impossible for the field to calculate the reliability of a system accurately.

The final configuration is submitted to Systems Assurance, as well as Systems Utilization, via a reliability request form. A study team also submits the reliability requirements of the system. Systems Assurance performs the reliability analysis and then lets the design team know whether or not the system has in fact met the reliability requirements.

Duplexing is a method of increasing units that will perform the same function in order to ensure reliability, but it is not necessary for a small CBS.

12. RSDP's (for information only)

Changes caused by the network and terminal design, the reliability analysis, and the back-up plan may create RSDP's.

Terminal specifications for the existing network and existing terminals can cause RSDP's to be initiated for the 2701, 2702 or 2703 -- for example, if the 2702 has to interface full-duplex lines or contention terminals, RSDP's must be initiated for special terminal controls.

Once it is discovered that an RSDP is necessary, it must be submitted. The requestor is advised to describe it hardware-wise and function-wise in as much detail as possible. This will enable the engineers designing the RSDP's to meet the requestor's specifications without waste of time and effort. Prior to submitting an RSDP, it is advisable for the requestors to find out if similar RSDP's have been initiated in the past and approved. This job is performed by the Special Equipment Engineering Department.

For a small CBS as defined in this Handbook, no T/P RSDP's are considered.

13. Final Configuration

Prior to entering this task, logical file organization, final terminal and network design, and core estimates must be completed. The reliability analysis, back-up plan, and RSDP's are tasks that can be performed concurrently, but must be completed prior to the end of the final configuration task. Here is where the equipment list, configuration, and price are produced.

14. Operating Plan

This task is performed in parallel with all other tasks. The functional specifications are used as the preliminary basis of this task. The operating plan describes the procedures to be followed when preparing information as input to the system at the various remote locations and at the central location. The operating plan also describes the procedures to be followed after the various outputs are received at the remote locations and at the center. It further describes how the total system, including customer operating personnel, common carrier equipment, terminals and central systems, constitutes a unit to accomplish the design functions.

15. Phase-In Plan

A phase-in plan is the approach for designing and installing the system in logical phases. The final configuration and functional specifications enable the design team to produce a phase-in plan. Because the system involves a network, the common carrier must be contacted to see if the phase-in plan and schedule are realistic and meet with the carrier's approval. This task consists of planning a systematic approach to the implementation of a communications-based system.

16. System Analysis (Refer to Subsection 3. 8)

The major analysis performed here is the throughput analysis. This task evaluates the total system to see if it meets the required specifications. There are various studies and analyses that must be performed on a system. The throughput analysis is made up of a channel utilization study, file utilization study, buffer analysis, and processing-time analysis. As mentioned previously, an analysis must be performed on the network. If any of these analyses or studies shows that the system does not meet the specifications, a redesign of the system must be initiated.

Various techniques and tools are offered that enable the system designer to perform a system analysis. These tools fall into two categories:

(1) Yellow-pad techniques for simple systems and/or small systems: This means analytical techniques used in evaluating a system. If the system is not too complex and the designer is versed in statistics and probability analysis and has reference to IBM analytical methods, he can then evaluate the system without the use of simulator programs. This Handbook offers simplified tools to the user to enable him to perform a system analysis on a small CBS.

(2) Simulation for complex systems, such as a GPSS and CSS and network design programs.

17. Rework

This task results from the system analysis when it is found that the system does not meet its specifications. It can consist of minor changes to the system design or an overall change.

18. System Evaluation

This task is a result of the final system design. The questions to be resolved are as follows:

(1) Is IBM competitive?

(2) Is the system economically sound? (cost justification)

(3) Should we continue with the system design or discontinue?

19. Lack of Interest

This is not a task, but is mentioned to emphasize that, after the system evaluation, it may be decided not to propose the system because the job is not feasible at the given point in time.

20. Preliminary Implementation Plan (Refer to Subsection 7. 0)

After it has been decided to propose the system, a preliminary implementation schedule has to be created. In turn, this preliminary plan affects the system evaluation as far as manpower and cost estimates for the implementation phase are concerned.

21. Proposal

This task is performed in parallel with the final tasks of the system design, but cannot be completed until the completion and evaluation of the total system design. The draft of the proposal must be completed prior to the Systems Assurance review.

22. Systems Assurance Review (Refer to Subsection 3. 12)

The review comes prior to submitting the proposal to the customer and after the system has been designed and the proposal written. It is advisable to have a preliminary review scheduled in case there is need to rework the design.

### 3. 1. 2 SYSTEM DESIGN CHECK LIST

A complete check list for system design follows. It is left to the discretion of the system designer whether or not all or part of the list should be considered. This list is for both small and complex systems and is an exhaustive list.

(1) Study customer problem

(2) Define scope of job

(3) Schedule

(4) Assign tasks

(5) Documentation plan

(6) Education

(7) Functional specifications

(8) Throughput objectives

(9) Data collection

(10) Data reduction

(11) Message types and formats

(12) Response requirements

(13) Traffic analysis

(14) Terminal RSDP

(15) Final terminal design

(16) Communications facility

(17) Final network design

(18) Displaceable cost

(19) Program design

(20) Core estimates

(21) Logical file organization

(22) Central RSDP's

(23) Reliability requirements

(24) Duplex philosophy

(25) Reliability analysis

(26) Back-up plan

(27) Final configuration

(28) Operating plan

(29) Update functional specifications

(30) System analysis

(31) Phase-in plan

(32) Common carrier meeting (PTT)

(33) Preliminary installation plan

(34) Rework, if necessary

(35) System evaluation

(36) Preparation of proposal

(37) Systems Assurance review

### 3.1.3 CONCLUSION

Let us now briefly review what has been covered in the design of a CBS system. We have seen the relationship of each major design task to the others, as well as a summary description of the individual tasks. In addition, emphasis has been placed on the areas that are unique to a CBS design, such as traffic analysis, system analysis, RSDP's, networks, and terminals. Manpower estimates have been made for the design phase. The point to be made here is that a CBS can be designed properly in a realistic time schedule when approached systematically. This is one of the chief purposes of this Handbook.

The next subsection will cover the development of the functional specifications in detail.

## 3. 2 FUNCTIONAL SPECIFICATIONS (Information Required from Customer)

This subsection points out the type of information required from the customer in order to develop the system's functional specifications. It is the purpose of this subsection to emphasize the fact that dynamic traffic statistics, in the form of messages in and out per required terminal location during the peak period, are one of the most important design parameters included in the functional specifications, along with the actual description of the required system functions. This section also shows the user some methods of defining, collecting, reducing, and analyzing dynamic data.

Refer to the 1050 Example, subsection 4. 5, for an example of functional specifications for a sales order entry system.

Before describing the information required as the basis of the system design, it is important at this point to define what a functional specification is.

A functional specification is a document that contains all the information required to design a communications-based system, such as:

1. Application description
2. Functions required of the system for message and line control
3. Functions required for applications communications programs
4. Terminal locations
5. Man-machine interface requirements and environment of operations
6. Suggested message formats
7. Traffic statistics
8. Throughput objectives
9. Response requirements
10. Reliability requirements
11. Type of error recovery required
12. Description of logical files
13. Implementation date and proposal date

### 3. 2. 1 APPLICATION DESCRIPTION

In the Application Description section of the Functional Specifications, the customer application is described -- i. e., sales order entry, reservation system, banking, insurance, credit rating, medical, etc.

The description of the application usually spells out the major operations in terms of how they are performed at present, such as editing, inventory control, methods of sending and receiving information, the flow of information from and to specific departments or organizations, billing, and updating of files.

Included in the application is usually a description of the functions of each of the personnel involved, such as clerks, order writers, supervisors, etc. The company's overall organization in terms of departments and responsibilities may also be included in the application description.

In addition, there may be a short description of other required applications, such as offline processing for payroll or the requirement of incorporating existing batch jobs with the communications-based system application.

### 3. 2. 2 FUNCTIONS REQUIRED OF THE SYSTEM FOR MESSAGE AND LINE CONTROL

In this part of the functional specifications is a description of all the functions concerned with the control of the network and traffic. This defines for the design team the type of hardware and programming combinations necessary to perform these functions. In a small CBS situation, it is up to IBM to ask the customer specific questions concerning the various message and line control functions in order to come up with this part of the specifications. The IBM team should describe each function to the customer and then discuss whether or not it is necessary for his application. This Handbook offers the following check list to the user as a basis for his questions to the customer.

Check List for Message and Line Control

1. Receiving and/or transmitting messages
2. Checking origination and/or destination codes
3. Routing messages based on destination codes

4. Detecting special characters, such as characters indicating End-of-Message and End-of-Transmission

5. Validity and format checking of information in the header of the message

6. Network awareness. Maintain the status of all lines and terminals

7. Sequence number checking of input messages and creating sequence numbers for output messages

8. Time and date stamping of messages

9. Keeping message statistics by line and terminal

10. Recognizing priorities and different message types

11. Speed conversion, taking a message from a line of one speed and placing it on a line of a different speed

12. Code conversion

13. Controlling network through polling and addressing

14. Providing management awareness of abnormal conditions

15. Responding to management corrective action

16. Keeping a historic log of input and/or output messages

17. Providing for message retransmission

18. Recognizing a failed terminal or line

19. Start-of-day procedures: good-morning messages

20. End-of-day procedures: good-night messages

### 3. 2. 3 FUNCTIONS REQUIRED FOR APPLICATION COMMUNICATIONS PROGRAMS

This part of the specifications elaborates on the application in detail and describes the functions required of the communications-based system necessary to operate the application. The basis for these functions is a definition of each transaction and the operation that must be performed on the transaction. The transaction description also includes what must result from the transaction.

This section also includes functional descriptions of such operations as file updating, necessity for multiprogramming both on- and offline programs, and spooling.

To aid the user in defining the transaction types, it is recommended that the documentation for this effort should list all the transaction or message types received by the center, along with the response expected from the system to the transaction. An example of this documentation is as follows:

| Input Transaction | Reaction |
|---|---|
| 1. Input Message (Administrative) | a. Verify origination<br>b. Log message<br>c. Check sequence number<br>d. Verify destination<br>e. Make appropriate translations<br>f. File in outgoing log |
| 2. Order | a. Adjust inventory<br>b. Log<br>c. Produce shipping notice<br>d. Produce picking tags to identify items |
| 3. Customer Record Change | a. Log<br>b. Change record<br>c. Acknowledge |

### 3. 2. 4 TERMINAL LOCATIONS

Within this section of the specifications, there should be a map or other type of description designating the desired terminal locations and indicating one of them as the location for the CBS's center. The user is advised to obtain a map of the area or country and pinpoint the exact terminal locations. This is required for an accurate network design and cost estimate of the communications facilities.

### 3. 2. 5 MAN-MACHINE INTERFACE REQUIREMENTS AND ENVIRONMENT OF OPERATIONS

At this point of the functional specifications, it is important to indicate the customer's unique requirements as to hardware features or operations that will make the execution of his application as simple and straightforward as possible.

The information required from the customer would be:

1. Time period in hours per day and days per week for the communications-based application.

2. Time period in hours per day and days per week for the offline batch operation.

3. Indication of scheduled transmissions of messages or transactions between the terminal locations and the central.

4. Type of output document required:

   a. Cards
   b. Printed page
   c. Display of some type

5. Type of input method required:

   a. Cards
   b. Keyboard online
   c. Keying with verification
   d. Buffer requirement
   e. Badge

6. Personnel to operate the new terminals:

   a. Managers
   b. Experienced teletype operators
   c. Experienced typists
   d. Kcypunch operators
   e. Insurance agent
   f. Engineer

7. Actual physical location of terminals

   a. Office
   b. Bank
   c. Railroad yard
   d. Factory

### 3. 2. 6 SUGGESTED MESSAGE FORMATS

A format must be defined for each message type, both input to and output from the system. Each field of the message must be indicated, showing the average number of characters for each field of every message type. Generally speaking, the customer does not have an existing communications system. Therefore, the suggested message formats must be obtained from documents, such as orders, bills, tickets, or invoices. A suggested method of documenting these forms is shown in the following sample order.

ORDER FORMAT

| NAME OF FIELD | % OF TIME | CHARACTERS | |
|---|---|---|---|
| | FIELD PRESENT | MIN. | MAX. |
| TERMINAL ADDRESS | 100 | 2 | 2 |
| SEQUENCE NUMBER | 100 | 6 | 6 |
| MESSAGE TYPE | 100 | 1 | 1 |
| CUSTOMER NUMBER | 100 | 8 | 8 |
| ORDER NUMBER | 50 | 0 | 8 |
| SHIP-TO KEY | 100 | 1 | 1 |
| NUMBER OF ITEMS | 100 | 2 | 2 |
| ITEM NUMBER | 100 | 8 | 8 |
| QUANTITY | 100 | 1 | 3 |
| END OF MESSAGE | 100 | 2 | 2 |

To define message formats for inquiry requests and their answers (which do not exist in any form at the customer's location), it is necessary to list the type of information required by the customer in reply to the various inquiries. If the answer is in terms of categories, a good estimate of the average number of characters for each category can be made. The format of the inquiry request would be based on the type of information the customer receives in the answerback. (Refer to the 1050 Example, 4. 5, for typical inquiry formats.)

### 3. 2. 7 TRAFFIC STATISTICS

Traffic statistics for a system with a non-existent network are dependent on collecting dynamic data. This will define the number of input and output messages per terminal location during the peak time period and during the average day. It also defines the message mix during the peak time period, which is the percentage of each transaction type during this time. This part of the specifications lists all terminal locations with their associated traffic during the peak time period and average day. To obtain the relevant data, it is necessary to take either or both of the following approaches:

1. Review past records kept by the customer;

2. Take actual statistics of traffic at each proposed terminal location for a specific

period of time.

An example of the first approach for a sales order entry system is:

a. Data can be obtained from past records at the central office, such as:

   i. Records of total number of line items processed per location by past years and months. This helps to determine the peak month.

   ii. The number and type of orders processed are usually on record in terms of past monthly and yearly volumes per location. This also helps to determine the peak month.

b. Information can be obtained from the remote location managers, such as:

   i. Average number of items per order,

   ii. Average number of characters per item,

   iii. Peak time period for priority orders

c. Reduction of data for traffic analysis for each location could be as follows:

   i. Take peak month figures and reduce to the peak time period by dividing by hours per day and days per month, if appropriate, in terms of orders.

   ii. Multiply number of orders by the average number of characters per order.

   iii. Multiply orders by the average characters in the output document generated by the order.

   iv. Obtain the total traffic in and out by messages/hour and characters/second.

   v. Apply contingency factor for growth, error, and change in message format.

   vi. Add a percentage to the total characters per message for control characters (usually 10%).

If the second approach to taking actual data is used for arriving at the traffic statistics, data-taking forms are usually necessary, along with a data-taking form specification. The data-taking form specification describes the use of the forms, as shown in the following example.

Data-Taking Form Specification

1. All locations would submit all the information required on the forms to the system design group. The forms would be prepared on a daily basis from October 1 to October 30, with a summary.

2. The following is a description of each form:

   a. Form 1 - Daily Telephone Order Report (refer to Figure 15): Each order clerk should document the phone orders as they are received, by hour, each day. They would be in accordance with the code in the upper left corner of the form. The number of lines on the order would also be recorded.

   b. Form 2 - Daily Mail Order Report: Daily orders received by the location in the mail could be tabulated in a similar manner to the telephone orders, excluding the time indication.

After the data has been collected, the traffic must be analyzed to obtain the proper traffic statistics. The traffic analysis would produce information for each location, such as:

1. Message mix

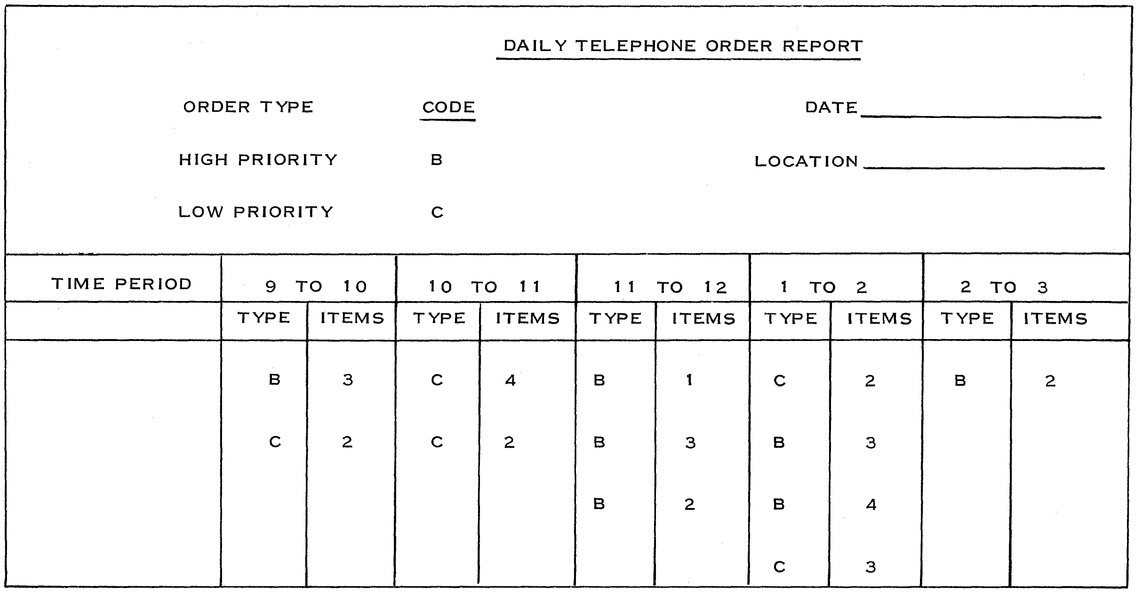
2. Input/output ratio

3. Total number of messages and order types by month, day, and hour

4. Average, maximum, and minimum number of lines per order type.

5. Average, maximum, and minimum number of orders per day

6. Peak day and hour for orders and line items



DAILY TELEPHONE ORDER REPORT

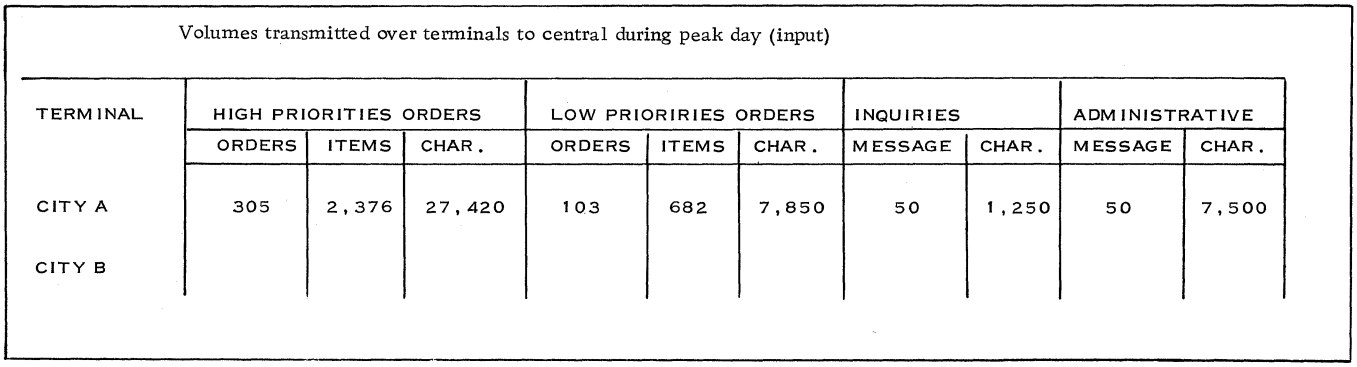
| ORDER TYPE | CODE | |
|---|---|---|
| HIGH PRIORITY | B | |
| LOW PRIORITY | C | |

DATE

LOCATION

| TIME PERIOD | 9 TO 10 | | 10 TO 11 | | 11 TO 12 | | 1 TO 2 | | 2 TO 3 | |
|---|---|---|---|---|---|---|---|---|---|---|
| | TYPE | ITEMS | TYPE | ITEMS | TYPE | ITEMS | TYPE | ITEMS | TYPE | ITEMS |
| | B | 3 | C | 4 | B | 1 | C | 2 | B | 2 |
| | C | 2 | C | 2 | B | 3 | B | 3 | | |
| | | | | | B | 2 | B | 4 | | |
| | | | | | | | C | 3 | | |

Figure 15. Daily Telephone Order Report



Volumes transmitted over terminals to central during peak day (input)

| TERMINAL | HIGH PRIORITIES ORDERS | | | LOW PRIORIRIES ORDERS | | | INQUIRIES | | ADMINISTRATIVE | |
|---|---|---|---|---|---|---|---|---|---|---|
| | ORDERS | ITEMS | CHAR. | ORDERS | ITEMS | CHAR. | MESSAGE | CHAR. | MESSAGE | CHAR. |
| CITY A | 305 | 2,376 | 27,420 | 103 | 682 | 7,850 | 50 | 1,250 | 50 | 7,500 |
| CITY B | | | | | | | | | | |

Figure 15A. Traffic Analysis Chart

The traffic analysis creates a chart (Figure 15A) that shows the relationship of all terminal points at the peak time period.

3. 2. 8 THROUGHPUT OBJECTIVES

The throughput objectives may be defined by saying that the customer wishes to handle the load as determined by the traffic analysis in a given period of time. Throughput objectives are stated in messages into and out of the central per peak time period.

3. 2. 9 RESPONSE REQUIREMENTS

Response time, as defined to the customer, is the delay between the time he places a message in the proper form for acceptance into the system and the time the result of the input message is available to the appropriate person and/or location.

A technical definition of "round-trip response time" is the total time required for a message whose first character is entered into a terminal queue until the last character is received at the receiving terminal.

Each message type must have a response requirement even if it is only in terms of days. For example, if a one-week response time is acceptable, it is possible to send the input by mail and have it processed at the computing center in a normal batch operation and then have the results mailed back to the originator. In this case, there is no need for a communications-based system. If a response requirement for input originating at remote locations is, on the average, one day or less, there is usually a need for a CBS.

Response time is a required parameter and defines basic information necessary for the terminal selection, communications facility selection, files, CPU, and programming approach.

For a small CBS, response requirements should not be rigid.

A form used to show the response requirements for each message type and the percentage of each message type that must meet the requirements is given in Figure 16.

### 3. 2. 10 RELIABILITY REQUIREMENTS

This portion of the functional specifications is for complex and/or duplex systems. It is not required for a simplex system such as the small CBS. (Refer to subsection 3. 3 if additional general information is required. )

### 3. 2. 11 TYPE OF ERROR RECOVERY REQUIRED

This section of the specifications indicates whether some of the following is necessary:

1. Message protection

2. File protection

3. Continuing a manual operation when the system fails

4. Detection of line, terminal, and front-end failures

### 3. 2. 12 DESCRIPTION OF LOGICAL FILES

A list of all the logical files necessary to implement message and line control and the communications application program are a necessary part of the functional specifications. In addition, statistical data must be collected for a valid estimate of how many records will be in each logical file. Standard methods of defining files for batch systems can be used.

The only logical files dependent on the dynamic data are those associated with line and message control, such as:

1. Log file: for the logging of input and/or output messages for the purposes of retrieval and summary statistics;

2. Queue file: for queuing of messages by line and/or terminal prior to transmission;

3. Intercept file: storing of messages assigned to terminals or lines that are temporarily down;

4. Checkpoint file: retention of intermittent status information to provide for a restart in the event of system failure.

### 3. 2. 13 IMPLEMENTATION DATE AND PROPOSAL DATE

The implementation date definitely affects the system design and should be included in the specifications. It defines the type of hardware and programming that can be incorporated in the design, based on their availability date.

The target proposal date is necessary, especially if competition is involved. It informs the design team how much time is available to cover all the tasks necessary for the system design so that they can plan accordingly.

### 3. 2. 14 CONCLUSION

The design team is responsible for making sure that all areas which will be the basis for the design have been covered; and the recommended way to do this is to prepare a document as suggested by this subsection of the Handbook. Now that the functional specifications have been defined, we can approach the actual detailed tasks and design techniques.

| MESSAGE TYPE | DESIRED RESPONSE REQUIREMENTS | MAXIMUM RESPONSE REQUIREMENTS |
|---|---|---|
| INQUIRY | 75% IN ONE MINUTE | 90% IN TWO MINUTES |
| ADMINISTRATIVE TO OTHER TERMINALS | 50% IN THREE MINUTES | 80% IN SEVEN MINUTES |

Figure 16. Form: Response Requirements for Message Types

## 3. 3 RELIABILITY

This subsection examines the factors relating to hardware reliability.

### 3. 3. 1 GENERAL PRINCIPLES

One of the prime differences between an ordinary DP system and a CBS is that the CBS is required to be available most of the time to accept data whenever it is transmitted, and/or answer inquiries whenever they are forwarded. The availability criteria can be met only through careful system design. Total system reliability depends on the reliability of the individual system components plus the equipment available for back-up, and the procedures for making the back-up equipment operational.

The subject of reliability is a sensitive one. For instance, IBM does not release reliability data to anyone, other than IBM personnel with a "need to know," without proper formal approval. Approval to release reliability data to a customer is granted only on an exception basis -- for example, for certain process control applications and for certain governmental agencies where limited reliability data is disclosed in a manner which does not discriminate among similarly situated customers.

Some of the reasons for this policy are:

1. System reliability information is expressed in reliability measures such as system availability, mean time to system interruption, and mean time to system restoration, which, although they sound simple, are highly complicated averages. Not only do they measure hardware characteristics, but also the adequacy of programming and operating procedures, the capability and training of the maintenance organization, the logistics of spare parts supply, and, in general, practically all aspects of using and servicing the equipment. Moreover, system reliability measures are obtained through probability calculations based on the reliability measures of individual machines, which, in turn, are projected from a combination of failure rates of the electronic components, analysis of machine structure, and related experience factors. Once these machines are installed and valid field experience is accumulated, the initial statistical projections should be modified accordingly. In view of this complexity of system reliability measures and of their derivation, reliability information is subject to misunderstanding and therefore to abuse.

2. The absence of industry standards on the definition, method of derivation, and interpretation of reliability measures prevents valid comparisons of reliability data and provides opportunity for inadvertent or deliberate mischief.

An understanding of these policy principles should assist the salesman and systems engineer in dealing with the customer's reliability questions and problems and in applying some of the practical policies and methods suggested below.

### 3. 3. 2 PRACTICAL POLICIES AND METHODS

Whenever he proposes a DP system including some T/P capability, the salesman or systems engineer must consider the reliability required for the installation:

1. The reliability which is necessitated by the application itself, including response requirements.

2. The minimum reliability the customer will tolerate in practice.

It is important to realize that the answers to these two points generally must be derived by the IBM sales and systems personnel themselves. To ask the customer what his application's reliability requirements are is usually not meaningful to him and his answer may cause the IBM personnel to design the system on the basis of incorrect criteria.

During system design, the systems engineer must plan on various techniques of minimizing the effects of equipment failures or failure of the equipment due to other sources, such as unavailability of power. (These techniques for a simplex small communications-based system are covered in subsections 3. 10 and 3. 11. )

It is noted that such system design techniques mentioned above do not require additional hardware. It is, of course, often desirable to add hardware and/or communications lines to strengthen an important and apparently unduly vulnerable system. Competent assistance should be sought to determine whether any additional components have to be added, based on the reliability requirements of the customer.

If the IBM personnel conclude they have no reliability problem, no additional components are necessary and they can directly undertake the systems assurance procedure described below.

If the IBM personnel conclude that they do have a reliability problem, they should apply for assistance to the country systems design or systems assurance function, as may be appropriate. When approaching the systems design or systems assurance function, they should always have the following information:

1. The reliability requirements of the system and the reasoning therefor;

2. An enumeration of all procedures and systems techniques planned which will minimize the effects of any failure;

3. An analysis of the hardware configuration currently planned, considering the effects of all possible failures and combinations of failures. As a result of this analysis, the IBM personnel should be able to state what component failures, including communications facilities, cause system interruption and the various modes of degraded operation;

4. The most dependable information available as to how much the customer can, and is prepared to, pay for the IBM system needed for added reliability.

Based on this information, systems design or systems assurance will be able to assist in selecting the most appropriate system configuration.

## 3. 4. QUEUING AND RESPONSE TIME

### 3. 4. 1 INTRODUCTION

Communications-based systems differ from traditional batch-type systems principally because of their ability to respond rapidly and process transactions with little or no delay, despite variations in input volume and timing.

The concept of communications-based system design is similar in many respects to batch applications, but with marked differences in emphasis. Simulation and queuing theory would be useful in both, for example, but these time-oriented techniques are much more significant in CB system design.

Although response time is a principal characteristic of CB systems, it does not always represent a clear demarcation between real-time and batch systems. Either type of system may be operated online and fully integrated with the mainstream of business. The essential distinction lies in how well inputs can be buffered. If inputs can be accumulated and sequenced in some designated order within a period of elapsed time (say several hours or a day), the system is fundamentally batch in nature; if inputs must be accepted and processed as they are received with no time delay, the system is operating in real time. In addition to responsiveness, then, another feature of a CB system is the unregulated order of the input.

When the time and sequence of inputs cannot be scheduled in a system and must be accepted randomly, their occurrence is described in terms of probabilities. In an economically feasible banking system, for example, where tellers must share a single file of all customer accounts, whenever several customers demand simultaneous service from the common facilities, queues will probably form, and some individuals will have to

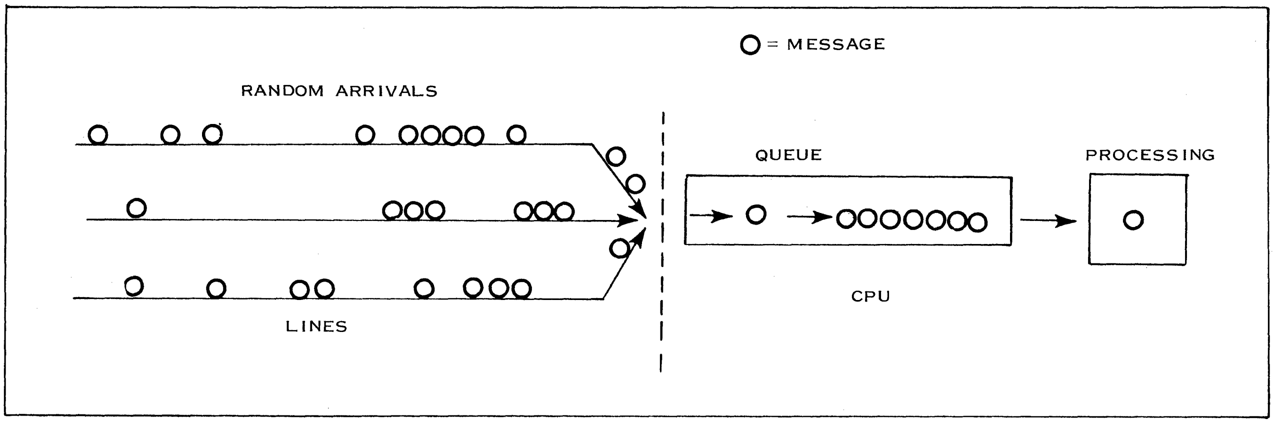


Figure 17. Distribution of Arrival and Service Times

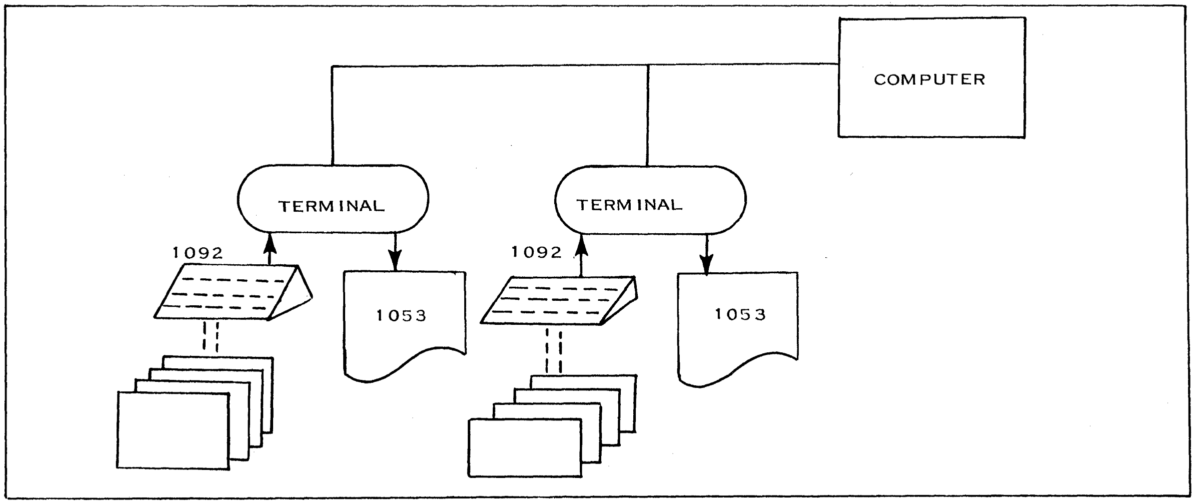


Figure 18. Multi-drop Network

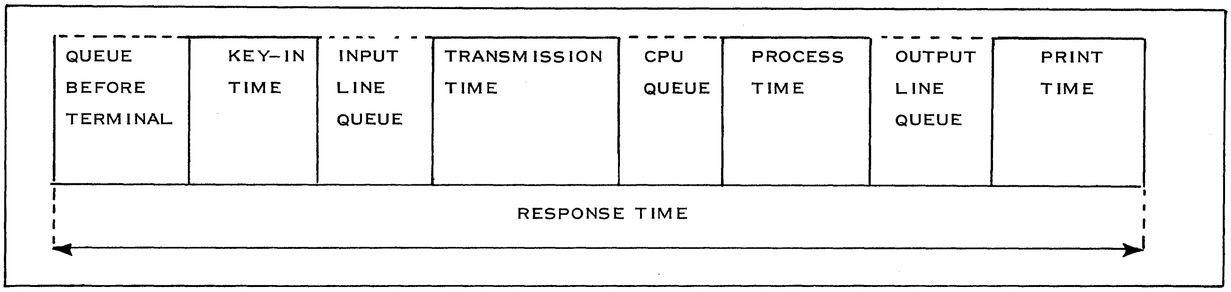


Figure 19. Queues and Service Times

wait before receiving service. It can be demonstrated mathematically that the amount of waiting time for a facility depends on several factors: arrival time and service rate, and the distribution of arrival and service times (see Figure 17).

In sum, we can state that in our system design, queues have to be considered from two aspects:

- queuing time, which influences the response time;
- queuing size, which influences storage requirements.

These considerations must rely mainly upon the worst conditions during peak activity.

If, for example, we consider a multi-drop network with terminals equipped with a 1092 programmed keyboard and a printer, we must consider the queues and service times mentioned in Figure 19.

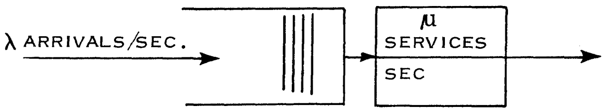
### 3. 4. 2 FUNDAMENTALS

Queuing theory studies analyze how a system behaves under statistically varying arrival and servicing situations: what idleness the service facility experiences, what congestion a long queue creates, or what form of the output distribution is.

If the arrival rate, $\lambda$ , is a great deal less than the service rate, $\mu$, the service facility stands idle much of the time (is little utilized) and the queue is of little or zero length. If $\lambda$ is only slightly under $\mu$, the queue is rather large and the service facility seldom idle (utilization high). If $\lambda$ is equal to or greater than $\mu$, the facility is always busy and the aueue becomes infinitely long as time goes on. In the rest of this subsection, $\lambda$ is always taken as less than $\mu$. The utilization coefficient --

$$\rho = \frac{\lambda}{\mu}$$

varies between 0 and 1, between a situation of no queue and minimum utilization of the facility, and a situation of a long waiting line and maximum utilization of the facility.



UTILIZATION COEFFICIENT $\rho = \frac{\lambda}{\mu}$

The general rule in such queuing situations is that the queue size and, therefore, the average delay exponentially increases with the utilization rate. This is shown in Figure 20.

So far, we have considered only the mean queue size and the mean queuing time. However, we may expect a certain deviation around this mean. For example, we might want to know the queue size and/or queuing time we may expect in 90% or 95% of the cases. This is called the "percentile".

From the considerations described above, we can conclude:

> A CB system cannot be dimensioned as a normal system for a utilization coefficient $\rho = 1$, because this would lead to infinite queues. Usually, to improve the total throughput, a system is not fully loaded by the CBS application, but runs simultaneously with a batch job.
>
> Practically, each required facility involved (i. e. , channels, access devices, CPU) should be designed in such a way that the utilization coefficient is much less than 1, and an advisable maximum would be 60%.

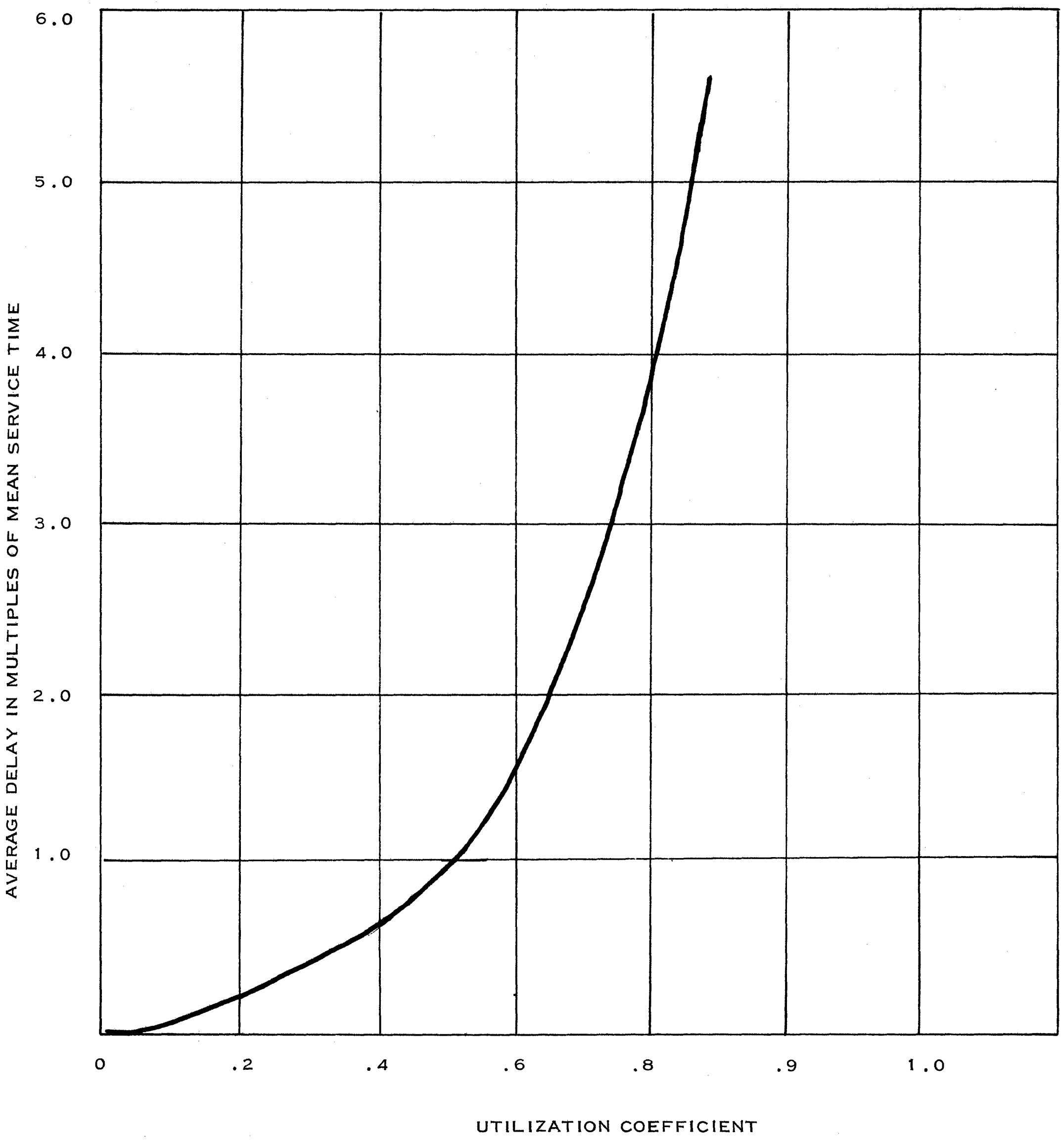


Figure 20. Average Delay for Random Service Times

From the utilization coefficient, we can also derive the maximum queue size we may expect in 90% or 95% of the cases. With these figures, we can determine the required buffer storage to achieve a certain service.

Similarly, we can find the mean queuing time and the maximum queuing time to be expected for 90% or 95% of the cases.

From the last figures, we can derive the average response time and the related percentiles.

### 3. 4. 3 CLASSIFICATION OF DISTRIBUTIONS

Without going into mathematical considerations, we must distinguish the various types of distributions involved in the processing of transactions in a CB system. The two extreme reference distributions against which comparisons are made are the exponential and the constant.

The exponential distribution is often called "random" because it describes a completely unordered process.

A constant distribution, on the other hand, describes a completely regular process in which there is no variation.

Most distributions of interest in CB systems have coefficients of variation inbetween the constant and exponential cases. These "inbetween" distributions are classified as "Erlang distributions with parameter m" or simply as "Erlang-m".

The Erlang parameter m is the measure of

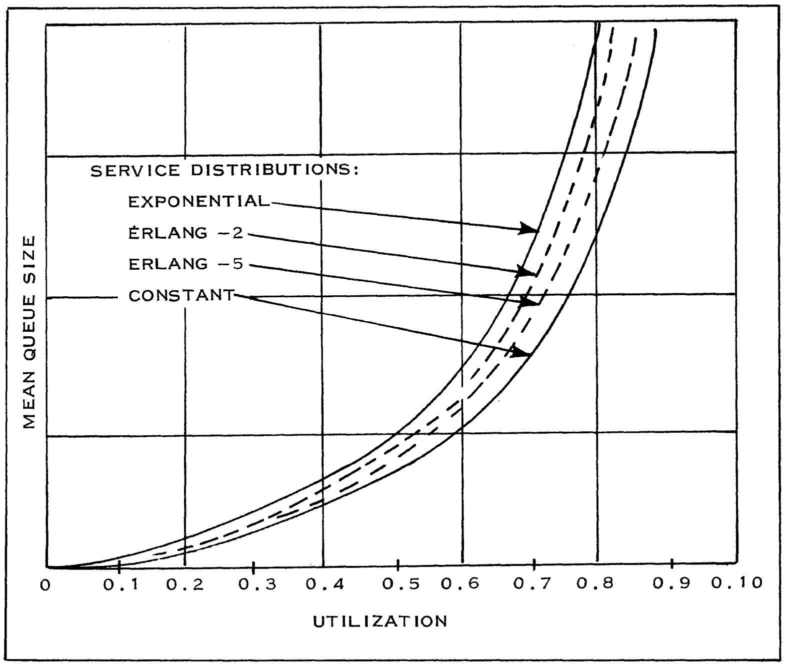
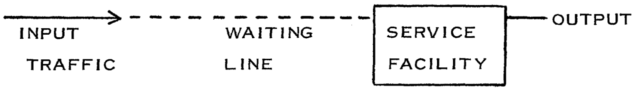


Figure 21. Queue Size vs. Utilization

randomness. The higher the value of m, the more constant the service time becomes (m = ∞ represents constant distribution); the closer m is to 1, the more random the distribution is. (m = 1 represents exponential distribution. ) (See Figure 21. )

A message or transaction process can be represented by:



CB systems can usually be considered as follows:

- Input Traffic

  Random distribution during the peak periods (m close to 1) unless we have specific knowledge about the input rate.

- Service Facility

  The distribution falls somewhere between constant and exponential, with some variation.

Usually, we do not know the distribution and we can only estimate it. But, as we are interested mainly in including a safety factor, it is advisable to take the upper (exponential) value. In this case, however, we are not far from the real situation. We can see in Figure 21 that the bounds for an exponential and constant distribution are close for a low-utilization coefficient (which is usually the case in a small CBS).

### 3. 4. 4 SOME SIMPLE FORMULAS

Without going into mathematical considerations, we present some simple formulas which can frequently be used to analyze a certain queuing situation. These formulas are based on the worst situation exponential arrival and exponential service distribution.

Queue Size:

The mean queue size, including the arrival which is serviced, now is:

$$\frac{\rho}{1 - \rho}$$

$\rho$ = utilization coefficient

Queuing Time:

The mean waiting time for service is:

$$\frac{\rho}{1 - \rho} \text{ x (service time)}$$

Figure 23 shows all the values of interest in analyzing a queuing situation for utilization coefficients up to 0. 6 (which, in general, we consider as an upper limit for a CB system). For more extensive information, refer to Analysis of Some Queuing Methods, Form F20-0007.

### 3. 4. 5 EXAMPLE

Multi-terminal Inquiry System

Suppose there are six inquiry terminals connected online to a computer. Processing time of an inquiry by the computer has a mean of 0. 5 seconds. The computer processes the inquiries one at a time, first-come, first-served.

Messages take 5 seconds to be keyed in and 2. 5 seconds to be printed out. The traffic during the peak period is 250 inquiries/hour. The traffic is the same for all terminals.

The terminal response time is defined here as the elapsed time for an inquiry entering the inquiry queue to the final printing of the answer and release of the terminal for the next inquiry. (See Figure 22. )

Per Terminal

We have a peak traffic of 250 inquiries/terminal/hour. Therefore, the average arrival rate will be:

$$\lambda = \frac{250 \text{ transactions}}{3600 \text{ seconds}} = 0.07 \text{ transactions/second}$$

Per inquiry, the terminal is occupied during the key-in time, process time, and print time.

This means: 5 + 0. 5 + 2. 5 = 8 seconds/transaction.

Therefore, one terminal can service:

$$\mu = \frac{1}{8} = 0.125 \text{ transactions/second}$$

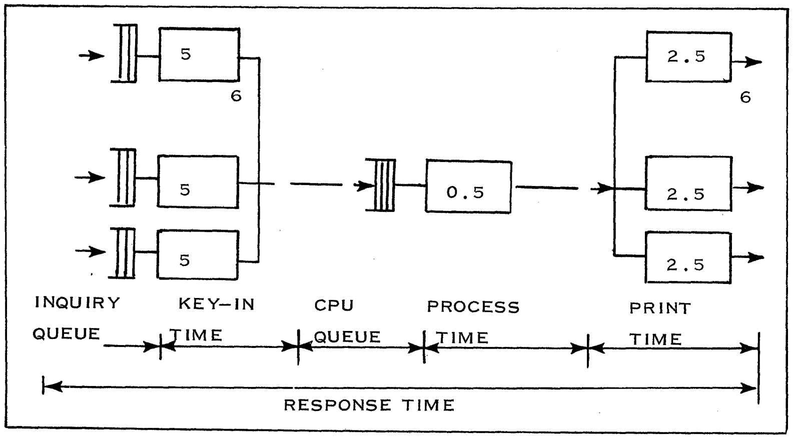


Figure 22. Terminal Response Time

This is the average service rate.

This means that the utilization coefficient of the terminal is:

$$\rho = \frac{\text{average arrival rate}}{\text{average service rate}} = \frac{\lambda}{\mu} = \frac{0.07}{0.125} = 0.6$$

We can use our table (Figure 23) to find the following conservative figures (for exponential service).

| | | | |
|---|---|---|---|
| - queue size | - mean | | 1.5 |
| | - percentile 90 | | 4.0 |
| | - percentile 95 | | 5.3 |
| - queuing time | - mean | | 2.6 x 8 sec. = 20.8 sec. |
| | - percentile 90 | = | 6.0 x 8 sec. = 48.0 sec. |
| | - percentile 95 | = | 7.8 x 8 sec. = 62.4 sec. |

Computer

There are six terminals, each with 250 inquiries/hour during the peak period, connected to the computer. This means that the average arrival rate for the computer will be:

$$\lambda = \frac{6 \text{ x } 250 \text{ transactions}}{3600 \text{ seconds}} = \frac{1500}{3600} = 0.5 \text{ transactions/sec.}$$

The processing time per transaction is 0. 5 seconds.

herefore, the computer has an average service rate:

$$\mu = \frac{1}{0.5} = 2 \text{ transactions/ second}$$

The utilization coefficient is:

$$\rho = \frac{\lambda}{\mu} = \frac{0.5}{2} = 0.3$$

Now, we can find in our table the following conservative figures (for exponential service):

| | | | |
|---|---|---|---|
| - queue size | - mean | | 0.4 |
| | - percentile 90 | | ≤1.5 |
| | - percentile 95 | | ≤2.1 |
| - queuing time | - mean | 1.5 x 0.5 = | 0.75 |
| | - percentile 90 = | 3.5 x 0.5 = | 1.75 seconds |
| | - percentile 95 = | 4.5 x 0.5= | 2.25 seconds |

From the above figures, we can conclude that for CB systems waiting delay at the terminal due to the terminal utilization coefficient has much more effect on the increase in response time than the waiting delay in the computer itself.

The CPU being so much faster than the rest of the system, it must operate at very high utilization before it causes waiting delays that are significant compared with other system delays.

Besides the queue locations mentioned in the example and in the figures, most of the queuing within the CPU itself is caused by file requests. When in a sequential processing system a disk file request for data or a program has been issued, no further processing occurs and the message is placed in the wait area. When both the channel and required disk file are free, a seek command is issued to position the arm and then a read or write command is given to bring in the data.

This queuing is included in the file response time, described more extensively in File Organization, subsection 3. 8. Moreover, we can derive the necessary figures from the curves shown in Performance Curves of IBM 2311 Disk Storage Systems, Form Z20-0309.

| UTILIZATION RATE | 0.1 | | 0.2 | | 0.3 | | 0.4 | | 0.5 | | 0.6 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SERVICE DISTRIBUTION | C | E | C | E | C | E | C | E | C | E | C | E |
| QUEUE SIZE | | | | | | | | | | | | |
| • MEAN | 0.1 | 0.1 | 0.2 | 0.3 | 0.3 | 0.4 | 0.5 | 0.7 | 0.7 | 1.0 | 1.1 | 1.5 |
| • STANDARD DEVIATION | 0.3 | 0.3 | 0.5 | 0.6 | 0.6 | 0.8 | 0.7 | 1.1 | 0.9 | 1.4 | 1.2 | 1.9 |
| • PERCENTILE 90 | 0.5 | 0.5 | 0.9 | 1.1 | 1.1 | 1.5 | 1.4 | 2.2 | 1.9 | 2.8 | 2.7 | 4.0 |
| • PERCENTILE 95 | 0.7 | 0.7 | 1.3 | 1.5 | 1.5 | 2.1 | 1.9 | 3.0 | 2.5 | 3.8 | 3.6 | 5.3 |
| QUEUING TIME | | | | | | | | | | | | |
| • MEAN | 1.1 | 1.2 | 1.1 | 1.3 | 1.2 | 1.5 | 1.3 | 1.8 | 1.6 | 2.1 | 1.8 | 2.6 |
| • STANDARD DEVIATION | 0.2 | 1.2 | 0.4 | 1.3 | 0.5 | 1.5 | 0.6 | 1.8 | 0.8 | 2.1 | 1.1 | 2.6 |
| • PERCENTILE 90 | 1.4 | 2.8 | 1.6 | 3.4 | 1.9 | 3.5 | 2.1 | 4.2 | 2.7 | 4.9 | 3.2 | 6.0 |
| • PERCENTILE 95 | 1.5 | 3.6 | 1.9 | 3.9 | 2.2 | 4.5 | 2.5 | 5.4 | 3.2 | 6.3 | 4.0 | 7.8 |

C = CONSTANT DISTRIBUTION
E = EXPONENTIAL DISTRIBUTION

Figure 23. Queue Size and Queuing Time vs. Utilization Rate and Distribution

## 3. 5 TRAFFIC ANALYSIS AND NETWORK DESIGN

### 3. 5. 1 NETWORK CONFIGURATIONS

1. Point-to-Point

There is one device -- either a terminal, a control unit, or a concentrator -- per line. This can be a leased line or a dial-up connection.

Note: A dial-up connection is, functionally, a multi-point configuration but, technically, a point-to-point connection. For this type of configuration we need no network design at all.

2. Multi-drop

There is more than one device -- either a terminal, a control unit or a concentrator -- connected to the same line.

This technique has the advantages of:

a. Lower line cost

b. Less expensive multiplexor (more terminals use one connection point)

The disadvantage is that response times are higher.

Note: When shared line adapters are used, this gives a maximum of four independent transmission channels per line.

### 3. 5. 2 LINE LOAD

The number of devices that can be connected to one line depends on:

- load of the different devices
- response time which can be accepted
- location of the devices

Normally, the design of a multi-drop network is based on line utilization or line loading.

Line load can be defined as the ratio of time used for handling transactions to the time available. The closer this ratio is to 100%, the longer the average response time, which is the total time a message is in transit, including all queues. This effect is explained in subsection 3. 4.

We start by calculating the line-holding time. Line-holding time is defined as the time the line is occupied for the handling of a transaction (including the single-poll or addressing time used to control that transaction). Line-holding time also includes such fixed functions as end-of-block, end-of-message, and line turn-around time.

For example, in conversational mode, the total line-holding time is the addition of individual holding times for input and output messages, plus internal processing time.

In fact, there is an additional line load for unsuccessful polls. However, when we have a line load of more than 60% due to actual message traffic, we can disregard the effect of negative polling on the line load. Moreover, in practice, we can organize our polling scheme in such a way as to have as few negative polls as possible.

We will derive a simplified procedure for calculating the response time as a function of the line-holding time and the line load for a multi-drop line.

The general formula for the mean number of messages waiting for completely random line-holding times is:

$$Lw = \frac{\rho}{1 - \rho}$$

$\rho$ = line load

Therefore, the mean waiting time due to queuing is:

$$Tw = \frac{\rho}{1 - \rho} \quad x \quad h$$

where "h" is the line-holding time per transaction.

The total response time in this case is the addition of the actual line-holding time and waiting time caused by the queuing at the devices:

$$R = Tw + h = \left(\frac{\rho \times h}{1 - \rho}\right) + h = \frac{h}{1 - \rho}$$

In an application where we have one transaction type with constant line-holding time, we can use the formula:

$$Lw = \frac{\rho}{2 (1 - \rho)}$$

The total response time is:

$$R = Lw \times h + h = \frac{\rho \times h}{2 (1 - \rho)} + h = h \times \frac{2 - \rho}{2 (1 - \rho)}$$

As a rule, there is an exponential increase in response time as a function of a linear increase in the line load.

For more detailed information, refer to Use of Communication Facilities in the Design and Implementation of Tele-processing Systems, Form Z20-1700, and the examples given in the Analysis of Some Queuing Models in Real-Time Systems, Form F20-0007.

### 3. 5. 3 NETWORK DESIGN

For the design of a multi-drop network, we can use the following instructions:

1. Plot the location of every terminal on a map.

2. Plot the location of the central processing center.

3. Plot the line load percentages per terminal beside each terminal location.

4. Establish the maximum allowable line loading factor (which depends on response time requirements).

5. If there are locations which exceed the maximum line load, divide each location into a number of maximum load lines, with one terminal for the remaining load.

6. Beginning at a terminal farthest from the central, link terminals together by straight lines with these objectives:

   a. Get as close to the maximum line load as possible (maybe slightly over it); then connect to the center and start a new line.

   b. Keep the line mileage to a minimum by pointing the line towards the central with narrow lateral detours to pick-up stations (see Figure 24).

   c. Any three adjacent stations to be linked are connected by the two shortest of the three possible sides of the defined triangle. This will occasionally result in branching. Branches may be of any length and the number of links may repeatedly branch.

   (Step 5 results in assigning terminals to a multi-drop line and connecting them together by a "minimum tree". This is the method used by PTT's to determine the mileage to be charged, regardless of actual wire routes.)

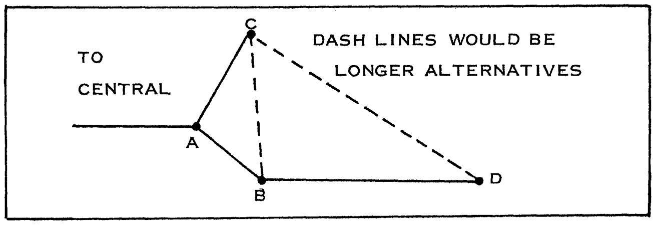
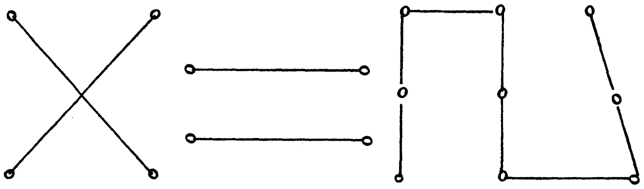


Figure 24. Location of Lines

   d. When several lines in the same part of the map have been completed, review them to try different combinations that will improve the objectives in step 5 above. Terminals in, or very near to, the central location can be selectively put on any line to raise its load without increasing it significantly. Try to avoid making connections which look like this:



   as they generally may be designed for a lower line cost.

7. When a satisfactory group of lines has been drawn, the cost of the network can be estimated by using PTT tariffs.

Very often, the optimum network design can be obtained only after consultation with the local PTT.

## 3. 6 HARDWARE OPERATIONS AND CONSIDERATIONS

Subsection 2. 2 roughly identified the functions of each piece of hardware and subsection 2. 5 gave a short functional description of the transmission control units. This subsection covers the transmission control units in greater detail.

The IBM 2701 and 2702 transmission control units and data are here the subject of a special study from a functional point of view. As this study includes the channel, we will first examine how a channel operates.

### 3. 6. 1 MPX CHANNEL OPERATION

Channel operation (summarized in Figure 25) is described as follows:

1. The CPU program initiates I/O operations with the start I/O instruction.

2. This instruction identifies the subchannel and the device (or the line) and causes the channel to fetch the CAW (Channel Address Word) from a fixed location in main storage.

3. The CAW designates the location in main storage from which the channel subsequently fetches the first CCW (Channel Command Word). The CCW specifies the command to be executed and, if any, the storage area to be used.

4. The channel issues orders to the device, stealing machine cycles for data transfer and obtaining new CCW's.

5. Termination of the I/O operation is normally indicated by two conditions: channel end and device

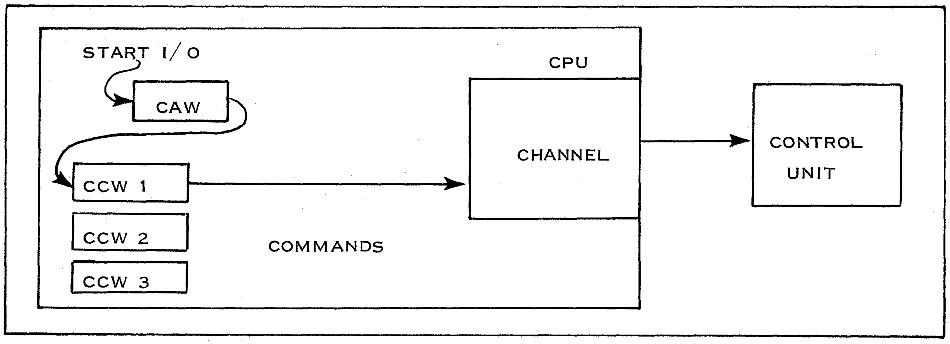
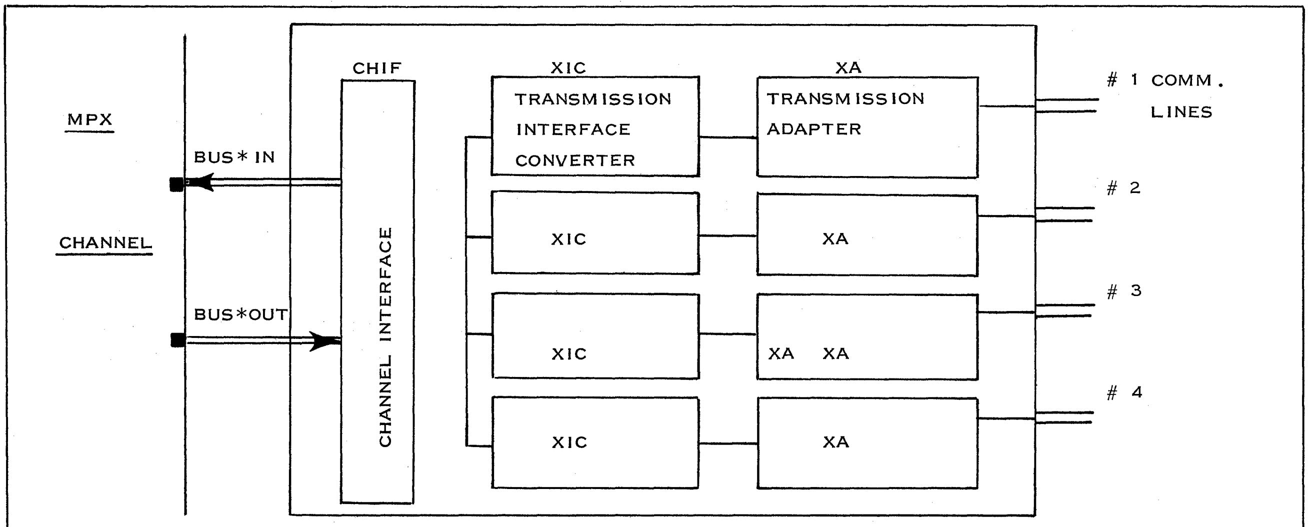


Figure 25. Channel Operation



Notes: A bus is a set of nine lines (8 bits plus parity) permitting access to the channel.
Bus-in is used to transmit address, status, sense information, and data to the channel.
Bus-out is used to transmit addresses, command control orders, and data to the control unit.

Figure 26. Internal Organization of a 2701 Data Adapter Unit

end. The first signifies that the channel has received or provided all information necessitated by the operation and no longer needs channel facilities. The second indicates that the device has terminated the operation. Those two conditions may occur simultaneously.

6. The conditions indicating the end of I/O operation can be brought to the attention of the program by I/O interruptions or, if the channel is masked, by a programmed interrogation of the I/O device.

For further information, the reader is referred to: IBM System/360 Principles of Operation, Form A22-6821, and IBM System/360 I/O Interface - Channel to Control Unit, OEMI, Form A22-6843.

The transfer of information between the 2701 and the I/O channel may occur in any of three permissible modes: byte mode, multiple-byte mode, and burst mode. The use of these modes is subject to some limitations. On the selector channel, the 2701 always operates in burst mode; on the MPX channel, the 2701 may operate in the three modes. This depends on the following factors:

- A switch on the 2701 itself permits operation either in byte of multi-byte mode.

- An adapter which is not buffered must operate in byte mode.

- Buffered adapters may operate either in multiple-byte mode or burst mode, but this latter operation should be used with the greatest caution because of CPU interference caused by it.

### 3. 6. 2 2701 DATA ADAPTER UNIT

The 2701 permits communication lines to be attached to a System/360 either via a selector channel or an MPX channel.

The 2701 data adapter serves as a control unit to the System/360 and is programmed in the same way. It can handle either up to four lines operating at 200 bps or 600 bps in start-stop mode, or up to two lines operating at 1200 bps in start-stop mode (2260), or 1200, 2000, 2400, 40,800 bps in synchronous (STR) mode. (See Configurators at the end of this Section.)

The 2701 is divided, functionally, in three parts (see Figure 26):

1. Channel Interface (CHIF)

   The CHIF connects the 2701 to the channel. The CHIF also selects the proper XIC (see below) and controls operation of the meter. There is only one CHIF per 2701.

2. Transmission Interface Converter (XIC):
   The XIC operates with the I/O interface, stores the status, sense and command bytes, decodes the addresses, develops and checks the parity of the data, and operates with the XA (see below).

3. Transmission Adapter (XA):
   The XA performs the functions associated with a given terminal or class of terminals, such as interface control, parity check, character recognition, data buffering and status, and sense byte generation.

The maximum of XIC-XA pairs that a 2701 can use is four, depending on the category of terminal adapter used. (See Configurators, subsection 3. 6. 4.)

Data Flow (See Figure 27)

To explain data flow, we will use as an example the operation of a start I/O instruction with a read or write command. The circled numbers in the following paragraphs correspond to the numbers indicated on Figure 27.

1. After a start I/O instruction is initiated by the program, a command byte present on the bus-out is transferred from the channel to the one addressed XIC-XA pair. This byte is stored in the command register ①. Receipt and transfer of the command byte are controlled by control lines and circuits connected to the MPX channel ②.

2. The command byte is then decoded. A command which is successfully decoded is executed. A command that is not successfully decoded is rejected by setting the unit check bit in the status byte.

3. The status byte is returned to the channel through the general control area ③ and the status control. The status byte stored in status register ④ is sent to the bus-in of the MPX channel in response to a command selection and indicates the condition of the 2701 (busy or not) and the acceptance or rejection of the command. If the status byte in reply to a read or write command contains all zeros, the command is accepted by the XIC-XA. At that time, the start I/O is terminated and the channel itself

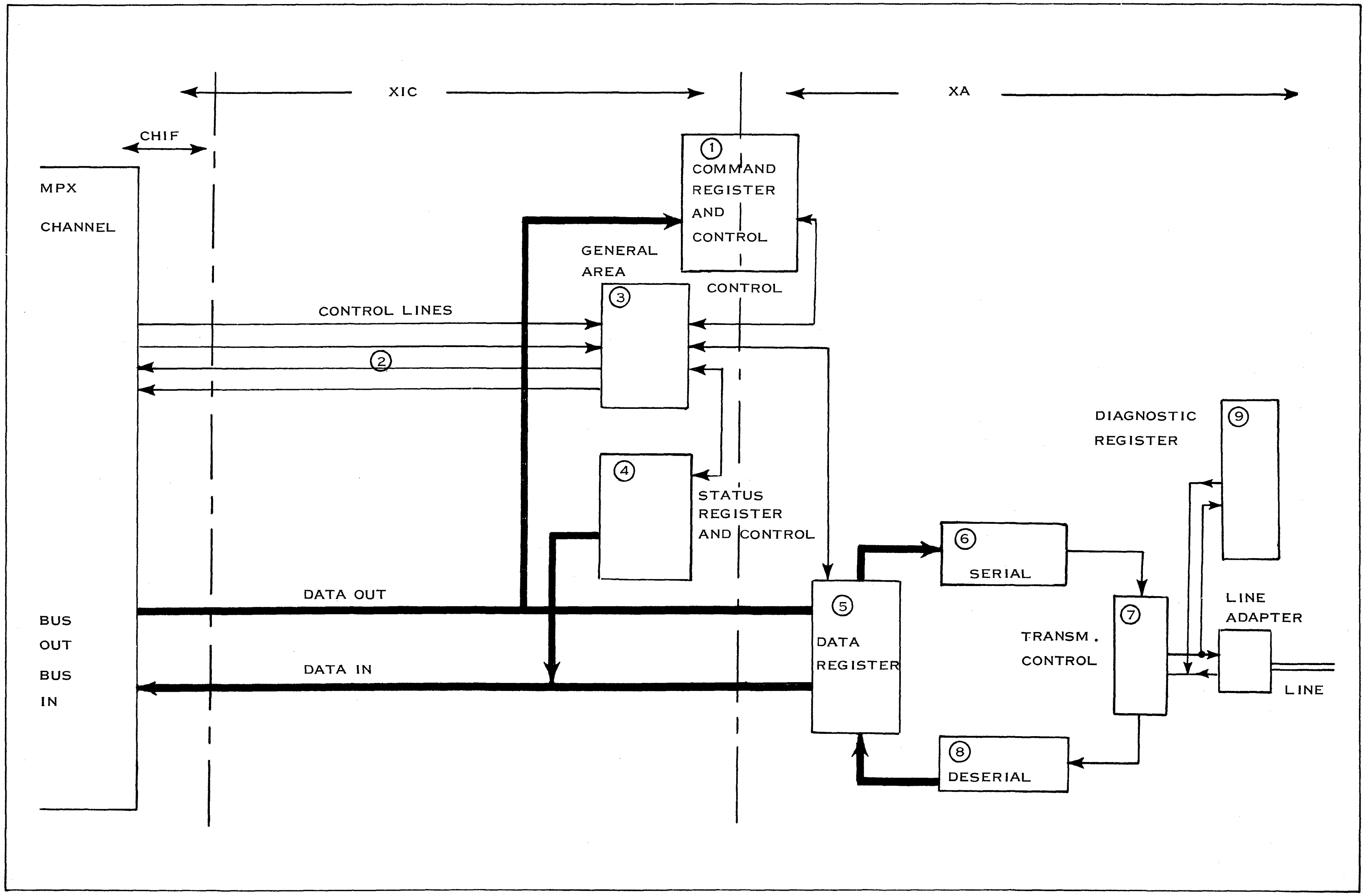


Figure 27. Data Flow

controls the following operations. The CPU will be interrupted either by the program (halt I/O, for example), at the end of channel operation, or if any abnormal condition occurs.

4. The data transfer sequence is performed after continual checking of the control lines between the channel and the 2701. The data is transferred either from bus-out to the data register ⑤ or from the data register to bus-in. Depending on the direction of the data flow, either the parity is checked or a parity bit is generated in the XIC.

5. If a write command is being executed, the XA serializes the character ⑥, inserts the start and stop bits or transforms the character by using the proper transmission code (depending on the XA type used), checks for special characters, accumulates and transmits the LRC character, where applicable. XMIT control ⑦ sets the modem in the proper transmitting condition.

6. If a read command is being executed, upon detection of a start bit the XA prepares to receive a character. The character which is received, bit after bit, is deserialized ⑧ and transmitted to the data register. The XA checks for special characters, checks for VRC, assembles LRC, where applicable, and requests data transfer to the bus-in of the channel.

7. At the completion of the last read or write operation, the status byte is sent to the channel with the condition "channel-end-device-end" and the channel requires an interrupt from the CPU. The operation can always be stopped at any time by the halt I/O instruction or by any abnormal condition detected by the channel or the control unit.

8. Diagnostic: A diagnostic register ⑨ is provided in the XA. This register holds the start bit, data bits, and stop bit. Through a diagnostic read command, the contents of this register are sent back to storage for checking.

9. Time-out: A 28-second time-out between data characters occurs when XA is operating on a read-type command. Time-out is started when the stop bit is received and stops when the next start bit is received. If the time-out elapses, the command is terminated and the channel is notified of this occurrence through the status byte. This prevents line or terminal failure from being undetected.

## Line Multiplexing

The selection of a line is made as follows:

Initial Selection Sequence:

1. The address of the 2701 control unit, together with the attached XIC-XA pair, is sent sequentially to all control units connected to the channel according to their priority.*

2. All control units compare this address with their own internally wired address. The 2701, which recognizes its own address, replies by sending its address back to the MPX channel.

3. At this time, the scanning performed by the MPX channel is stopped and the channel sends the command word to the selected XIC-XA pair.

4. The XIC-XA replies with the status byte and, upon acceptance of the command, the XIC-XA is now ready for the transfer of a data byte between the channel and the 2701.

Data Transfer:

5. The channel requests a transfer to or from the line.

6. If a read operation occurs, the XIC-XA replies by transferring its address with a data byte. If a write operation is performed, the channel transfers a data byte upon reception of the XIC-XA address. The transfer is achieved, byte by byte, performing operations 5 and 6 until the last character is received. (See Data Flow for the last operation.) After the last character is transmitted, the channel starts scanning again for another XIC-XA or control unit.

For further information, refer to: 2701 Data Adapter Unit, FEMI, Form 226-2018; 2701 Data Adapter Unit, Principles of Operation, Form A22-

*Each XIC-XA pair is identified by a unique I/O address. The priority of the XIC is determined only by its position within the 2701 and cannot be changed.

6864; and 2701 Data Adapter Unit, OEMI, Form A22-6844.

### 3. 6. 3 2702 MULTIPLEXOR

The 2702 permits the attachment to a System/360 of up to 31 lines operating at 200 bps, or up to 15 lines operating at 600 bps. (See Configurators, 3. 6. 4.)

The 2702 multiplexor serves the System/360 as a control unit. To the system program, each line attached to the 2702 is identified by a unique address and appears as a unique I/O device. All the lines attached to the 2702 are sequentially scanned by the 2702 itself (unlike the 2701, where line scanning is done by the channel). This operation will be described later on.

#### Hardware Description

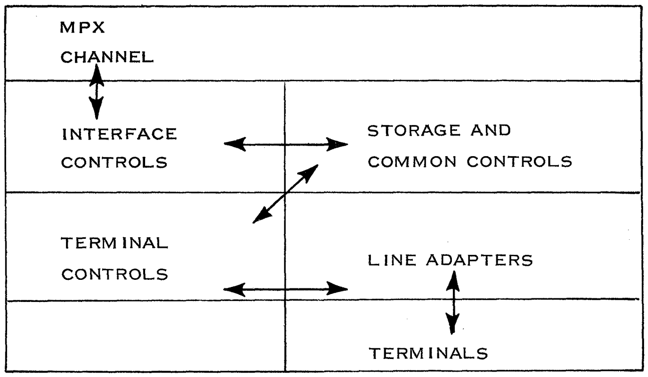


Figure 28. 2702 Functional Sections

The 2702 is composed of the following functional sections, shown in Figure 28.

1. Interface controls: These control all signal sequences with the MPX channel. Data transfer between the 2702 and the multiplexor channel is in byte interleaved mode.

2. Storage and common controls: These controls are common to all communication lines. They initiate the transfer of data bytes to or from storage. The Line Control Word (LCW) terminates channel commands by transferring the unit status byte to the channel, similar to the ending operation in the 2701.

3. Terminal Control: This control regulates the operation with the attached terminal, performing such functions as character recognition or checking of transmission code. These functions are characteristic of a given terminal type.

4. Line Adapters: Line adapters control the interface of communications facilities and provide bit buffering for the transmit data line. They permit either in-plant connection via units equipped with in-house modems, or attachment to IBM modems.

#### Functional Description and Data Flow

A 64-bit storage, called "line control word" (LCW) is provided for each attached line. It contains an assembly/disassembly area, a one-byte data buffer, a command area, LRC accumulation byte, and some status and control information for internal control. Actually, this storage is made up of two delay lines, each containing one half of the LCW (LCW 1 and LCW 2). During a read operation, the incoming bits are assembled in the LCW and, if one byte is complete, transferred to the one-byte data buffer. A flag bit is set to signal that service is required. The 2702 scans internally all LCW's and transfers the data byte from the data-byte buffer of the LCW into the I/O register of the 2702. If the MPX channel does not service the 2702 in time, internal scanning is delayed. If, due to this delay an LCW data-byte buffer is not emptied before the next byte is assembled (which is 1666 µs for a 600 bits/second line or 5000 µs for a 200 bits/second line), one byte is lost and this overrun condition is signaled by a special overrun bit in the LCW. The procedure between the 2702 and the MPX channel is the same as for the 2701: each data byte is followed by the line address, to allow multiplexing on the channel. Initialization of a read or write operation is the same as for the 2701.

Figure 29 shows that the 2702 can be considered to operate as a water wheel. The 2702 scans the lines continuously, stopping briefly at each line to collect bits to be put in unique locations or place them on the multiplexor channel. The "paddles" of the wheel are storage locations, or LCW's. The LCW contains an assembly/disassembly area, which permits the transformation from serial to parallel or vice versa.

The 2702 continuously scans the LCW's and carries data bytes, one at a time, either from the LCW's to the I/O register, or from the I/O register to the LCW's.

Two delay lines operating synchronously provide 64 bits of storage per line to make up the LCW for each line. One word is assigned to each communication line. The line control word contains an assembly/disassembly area (serial data), byte buffer (data buffer), command area, LRC accumulation, and status and control information for internal control.

The line control word interfaces with the MPX channel not only with the I/O register, but also with

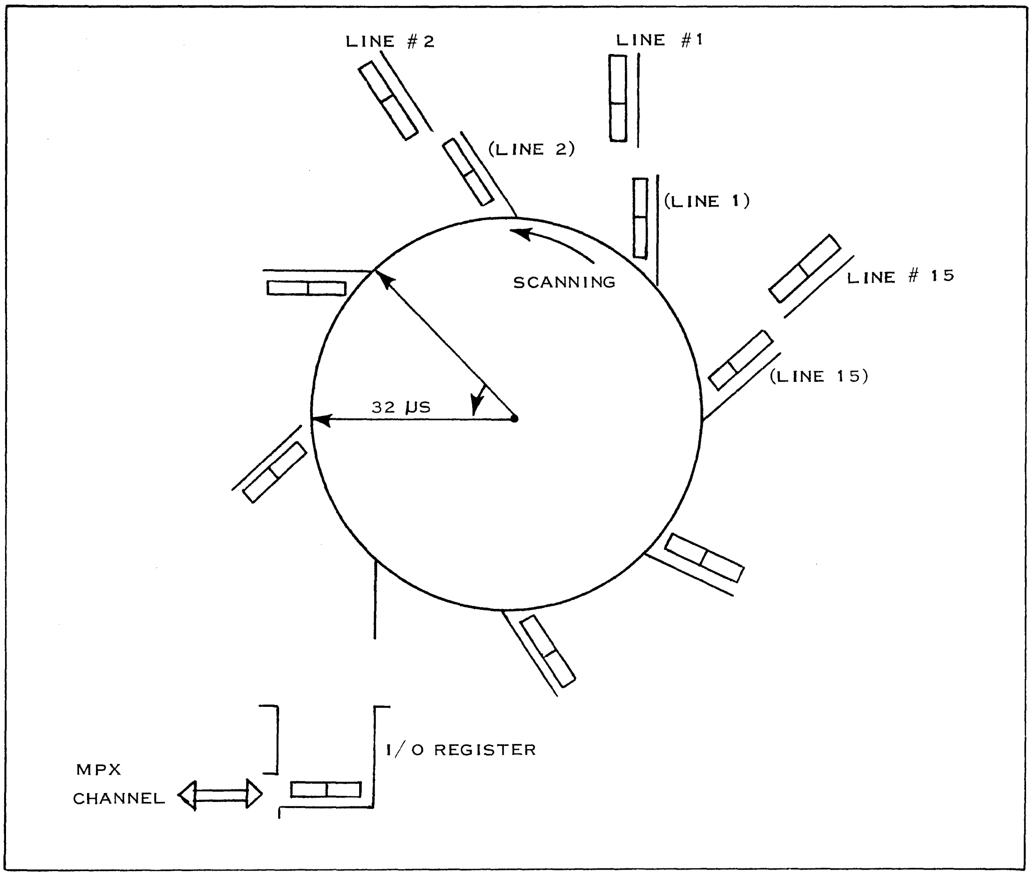


Figure 29. 2702 Schematic

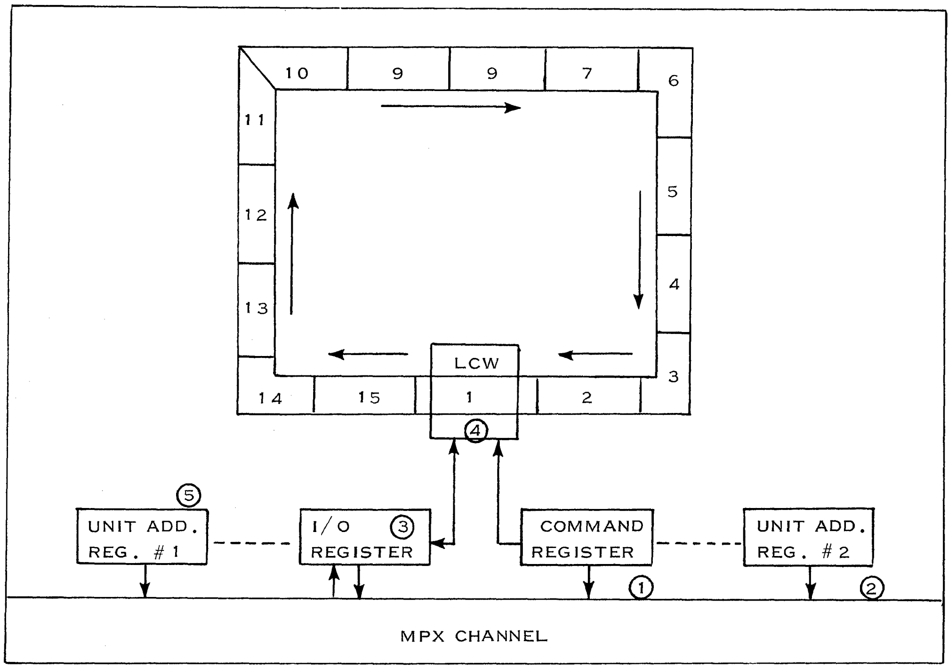


Figure 30. Address and Command Registers

the command register, address register 1, and address register 2.

Figure 30 clarifies the functions of the address and command registers. The circled numbers in the paragraph that follows correspond to the numbers indicated in Figure 30.

During initial selection, a command and line address are received by the 2702 and placed, respectively, in command register ① and address register ②. The status byte is sent back to the MPX channel (all zeros, if the command is valid). The I/O register ③ acts as a data buffer between the addressed LCW ④ and the bus-in or out of the MPX channel. Its contents are associated with the line control word designated by the address register 1 ⑤.

For a read command, the serial data field of the LCW assembles the incoming byte, bit by bit, from the line until a complete byte is received. The byte is then transferred to the data buffer field of the LCW and a special bit is set to indicate that one byte is assembled (character service). If this bit was already set (the previous byte was not serviced), another bit is set to indicate overrun condition. The time between two characters for 200 bps operation is 67.5 ms.

If the scan detects the character service bit set in the LCW and reset in both the I/O register and unit address register 1, then the data is transferred from the data buffer to the I/O register. Its reference line address is stored in address register 1.

Then the MPX channel serves the I/O register and transfers data and address to the bus-in of the channel. Until the I/O register is serviced by the MPX channel, the I/O register and unit address register 1 are not available for further service requests.

The following is a summary of all the operations performed during the execution of a command.

1. A command and line address are on the bus-out of the MPX channel. They are received by the 2702, checked for parity, and stored in command register and address register 2, respectively.

2. The address and command data are decoded and, following acceptance or not by the 2702, a reply is sent to the channel. If address and command are valid and accepted, an all-zero status byte is sent back to the MPX channel.

3. At the completion of the initial selection, the LCW referenced by address register 2 is accessed. The contents of the command register are loaded into the LCW as specified, into a special field in the command byte.

4. For a write command, data is transferred, byte by byte, from the channel bus-out and stored into the data buffer field of the LCW, then transferred to the serial data field of the LCW. Each bit is then sent to the line adapter, where it is buffered and transmitted to the line. This operation is performed under the control of the terminal adapter.

5. For a read command, the bits are received from the line and loaded into the serial data field of the LCW. Then the assembled byte is transferred to the data buffer field of the LCW and transferred to the bus-in of the channel after a parity bit has been added. During that time, the 2702 prepared to receive another character.

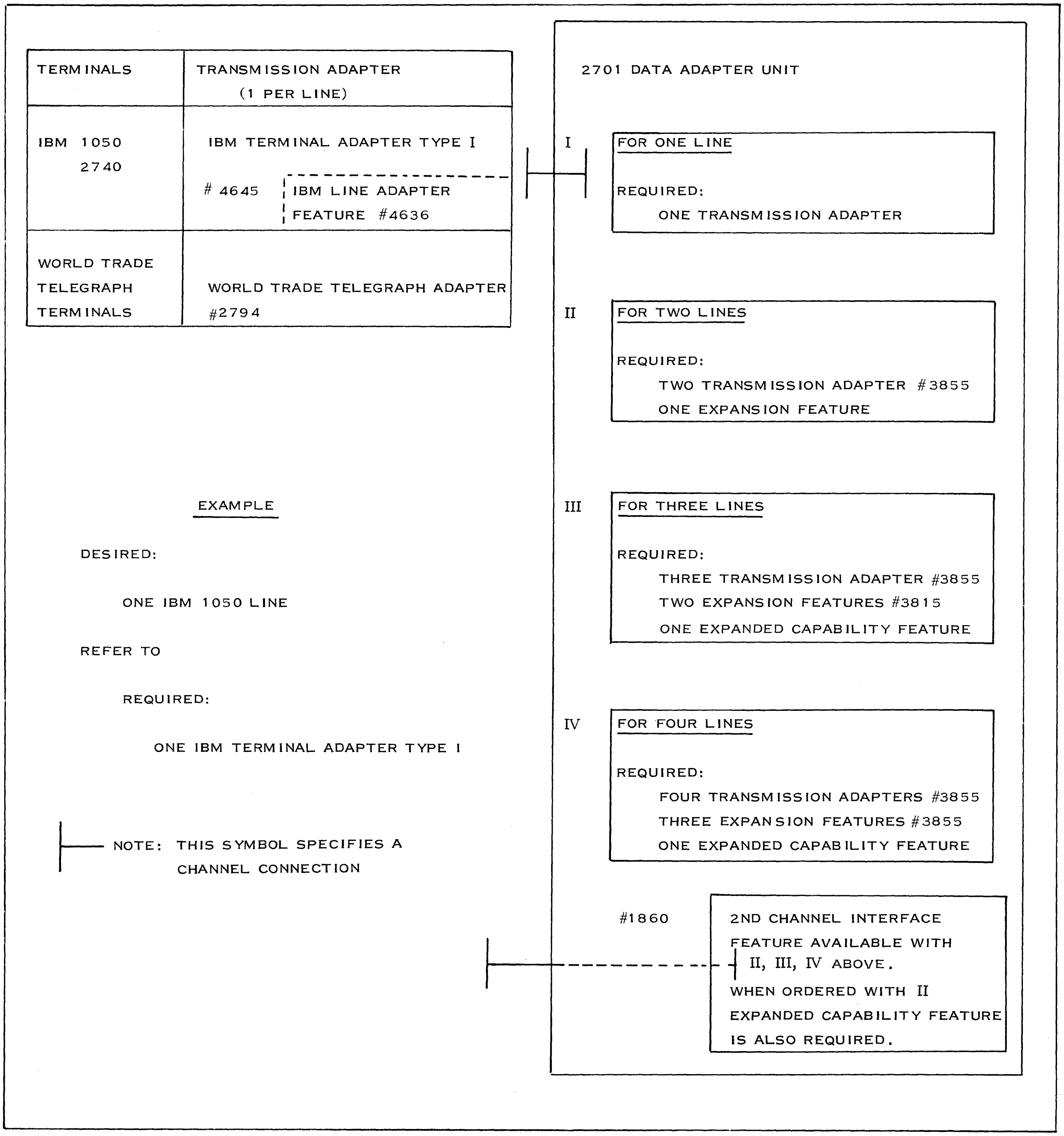
6. At the termination of command execution, a status byte is sent to the channel defining the conditions which the command ended. Normally, channel-end-device-end status is sent to the channel and requires interruption from the CPU.

For further information, refer to: FEMI - Theory of Operation, 2702, Form 226-2005; and System/360 Component Description, 2702, Form A22-6846.

### 3. 6. 4. CONFIGURATORS

The following are configurators for the IBM 2701, 2702, 1050, 2740, and 2260.

(Refer to the Data Communications Handbook for configurations of other terminals and multiplexors. This book is usually kept by the T/P Coordinator.)



| TERMINALS | TRANSMISSION ADAPTER (1 PER LINE) |
|---|---|
| IBM 1050<br>2740 | IBM TERMINAL ADAPTER TYPE I<br># 4645 IBM LINE ADAPTER FEATURE #4636 |
| WORLD TRADE TELEGRAPH TERMINALS | WORLD TRADE TELEGRAPH ADAPTER #2794 |

EXAMPLE

DESIRED:

ONE IBM 1050 LINE

REFER TO

REQUIRED:

ONE IBM TERMINAL ADAPTER TYPE I

NOTE: THIS SYMBOL SPECIFIES A CHANNEL CONNECTION

2701 DATA ADAPTER UNIT

I FOR ONE LINE

REQUIRED:
ONE TRANSMISSION ADAPTER

II FOR TWO LINES

REQUIRED:
TWO TRANSMISSION ADAPTER #3855
ONE EXPANSION FEATURE

III FOR THREE LINES

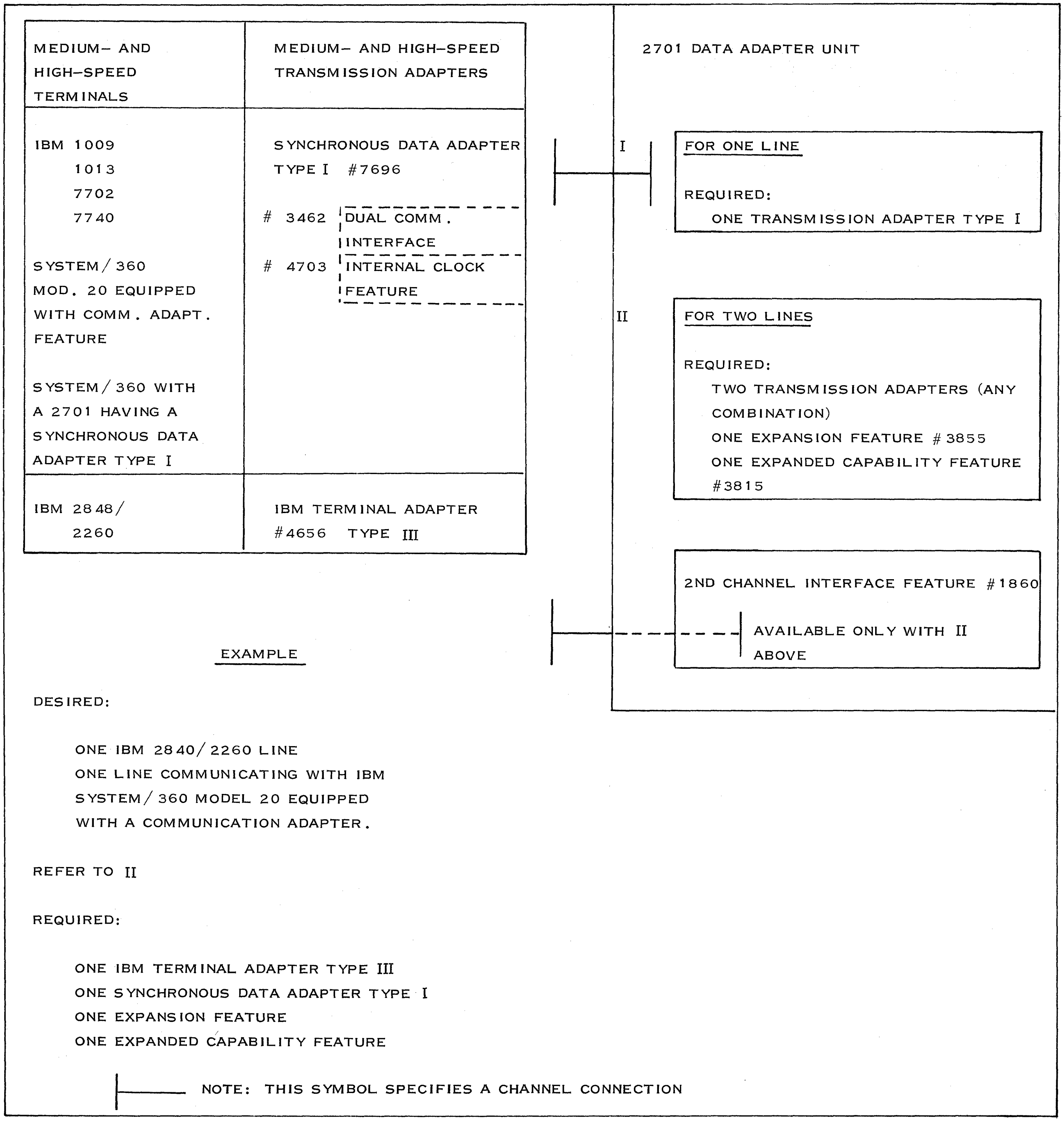
REQUIRED:
THREE TRANSMISSION ADAPTER #3855
TWO EXPANSION FEATURES #3815
ONE EXPANDED CAPABILITY FEATURE

IV FOR FOUR LINES

REQUIRED:
FOUR TRANSMISSION ADAPTERS #3855
THREE EXPANSION FEATURES #3855
ONE EXPANDED CAPABILITY FEATURE

#1860 2ND CHANNEL INTERFACE FEATURE AVAILABLE WITH II, III, IV ABOVE. WHEN ORDERED WITH II EXPANDED CAPABILITY FEATURE IS ALSO REQUIRED.

Figure 31. System/360 Data Communications and Acquisition Configurator for 2701 Low-Speed Communication Lines



| MEDIUM– AND HIGH–SPEED TERMINALS | MEDIUM– AND HIGH–SPEED TRANSMISSION ADAPTERS |
|---|---|
| IBM 1009<br>1013<br>7702<br>7740<br><br>SYSTEM/360 MOD. 20 EQUIPPED WITH COMM. ADAPT. FEATURE<br><br>SYSTEM/360 WITH A 2701 HAVING A SYNCHRONOUS DATA ADAPTER TYPE I | SYNCHRONOUS DATA ADAPTER TYPE I #7696<br># 3462 DUAL COMM. INTERFACE<br># 4703 INTERNAL CLOCK FEATURE |
| IBM 2848/<br>2260 | IBM TERMINAL ADAPTER #4656 TYPE III |

2701 DATA ADAPTER UNIT

I

FOR ONE LINE

REQUIRED:
ONE TRANSMISSION ADAPTER TYPE I

II

FOR TWO LINES

REQUIRED:
TWO TRANSMISSION ADAPTERS (ANY COMBINATION)
ONE EXPANSION FEATURE #3855
ONE EXPANDED CAPABILITY FEATURE #3815

2ND CHANNEL INTERFACE FEATURE #1860

AVAILABLE ONLY WITH II ABOVE

EXAMPLE

DESIRED:

ONE IBM 2840/2260 LINE
ONE LINE COMMUNICATING WITH IBM SYSTEM/360 MODEL 20 EQUIPPED WITH A COMMUNICATION ADAPTER.

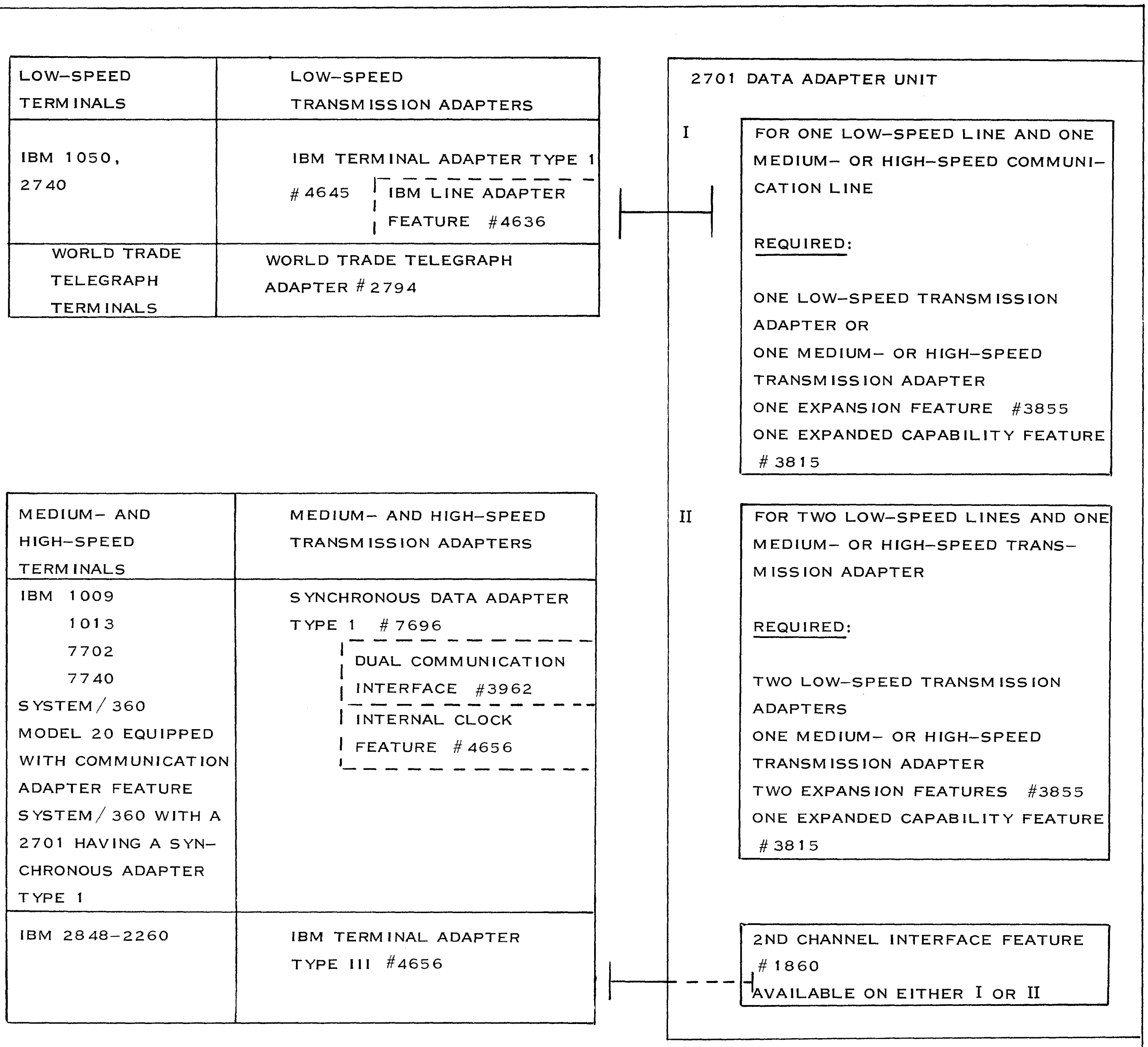
REFER TO II

REQUIRED:

ONE IBM TERMINAL ADAPTER TYPE III
ONE SYNCHRONOUS DATA ADAPTER TYPE I
ONE EXPANSION FEATURE
ONE EXPANDED CAPABILITY FEATURE

NOTE: THIS SYMBOL SPECIFIES A CHANNEL CONNECTION

Figure 32. System/360 Data Communications and Acquisition Configurator for 2701 Medium- and High-Speed Communication Lines



| LOW-SPEED TERMINALS | LOW-SPEED TRANSMISSION ADAPTERS |
|---|---|
| IBM 1050,<br>2740 | IBM TERMINAL ADAPTER TYPE 1<br>#4645<br>IBM LINE ADAPTER FEATURE #4636 |
| WORLD TRADE TELEGRAPH TERMINALS | WORLD TRADE TELEGRAPH ADAPTER #2794 |

| MEDIUM- AND HIGH-SPEED TERMINALS | MEDIUM- AND HIGH-SPEED TRANSMISSION ADAPTERS |
|---|---|
| IBM 1009<br>1013<br>7702<br>7740<br>SYSTEM/360 MODEL 20 EQUIPPED WITH COMMUNICATION ADAPTER FEATURE<br>SYSTEM/360 WITH A 2701 HAVING A SYNCHRONOUS ADAPTER TYPE 1 | SYNCHRONOUS DATA ADAPTER TYPE 1 #7696<br>DUAL COMMUNICATION INTERFACE #3962<br>INTERNAL CLOCK FEATURE #4656 |
| IBM 2848-2260 | IBM TERMINAL ADAPTER TYPE III #4656 |

2701 DATA ADAPTER UNIT

I

FOR ONE LOW-SPEED LINE AND ONE MEDIUM- OR HIGH-SPEED COMMUNICATION LINE

REQUIRED:

ONE LOW-SPEED TRANSMISSION ADAPTER OR
ONE MEDIUM- OR HIGH-SPEED TRANSMISSION ADAPTER
ONE EXPANSION FEATURE #3855
ONE EXPANDED CAPABILITY FEATURE #3815

II

FOR TWO LOW-SPEED LINES AND ONE MEDIUM- OR HIGH-SPEED TRANSMISSION ADAPTER

REQUIRED:

TWO LOW-SPEED TRANSMISSION ADAPTERS
ONE MEDIUM- OR HIGH-SPEED TRANSMISSION ADAPTER
TWO EXPANSION FEATURES #3855
ONE EXPANDED CAPABILITY FEATURE #3815

2ND CHANNEL INTERFACE FEATURE #1860
AVAILABLE ON EITHER I OR II

EXAMPLE:

DESIRED:

ONE IBM 1050 LINE, ONE SYSTEM/360 MODEL 20 LINE

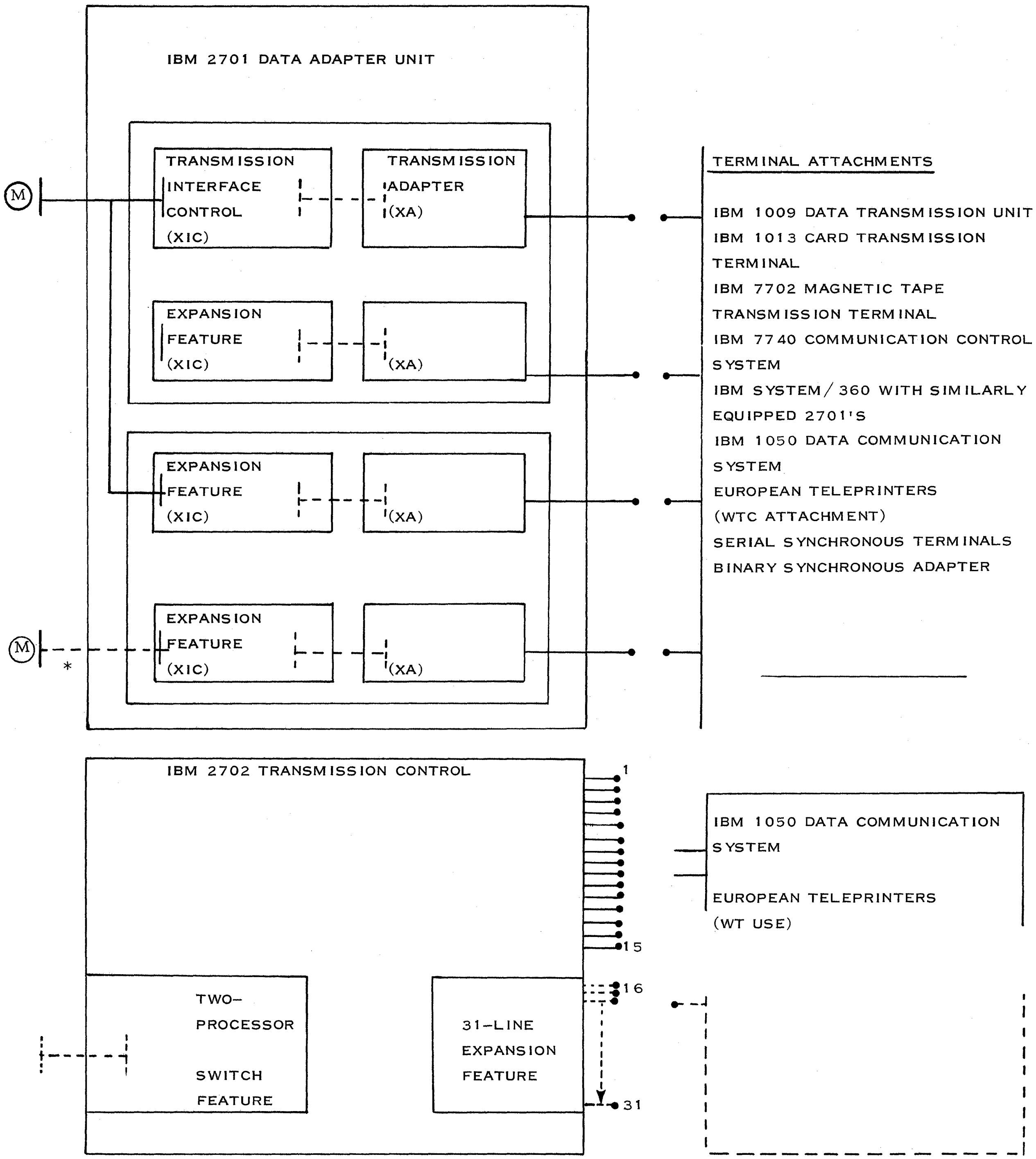
├── NOTE: THIS SYMBOL SPECIFIES A CHANNEL CONNECTION

REFER TO II

REQUIRED:

ONE IBM TERMINAL ADAPTER TYPE 1
ONE IBM SYNCHRONOUS DATA ADAPTER TYPE 1 WITH INTERNAL CLOCK FEATURE
TWO EXPANSION FEATURES
ONE EXPANDED CAPABILITY FEATURE

Figure 33. System/360 Data Communications and Acquisition
Configurator for Combinations of 2701 Low-, Medium-, and High-Speed Communication Lines

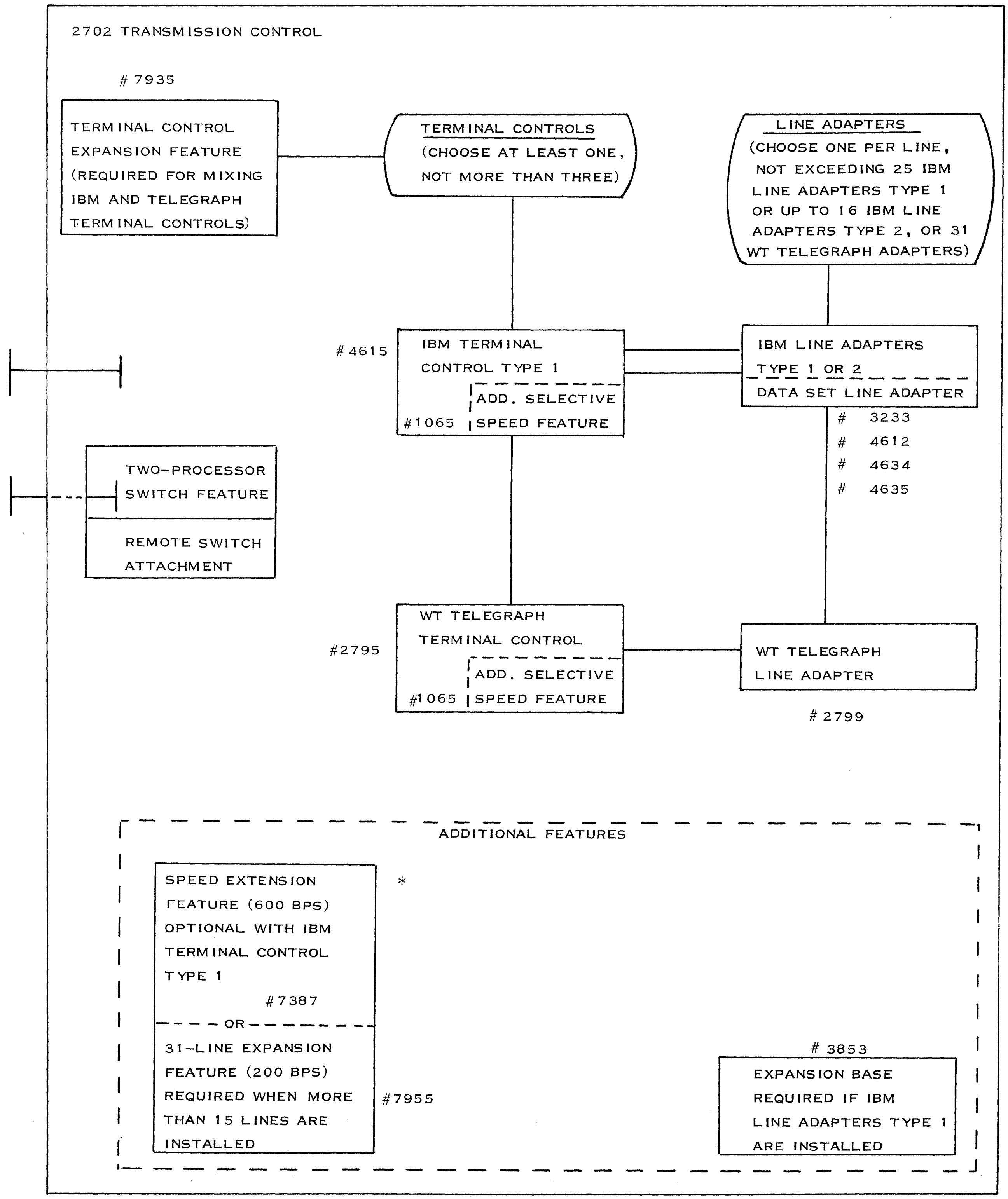

IBM 2701 DATA ADAPTER UNIT
TRANSMISSION INTERFACE CONTROL (XIC)
TRANSMISSION ADAPTER (XA)
EXPANSION FEATURE (XIC)
(XA)
EXPANSION FEATURE (XIC)
(XA)
EXPANSION FEATURE (XIC)
(XA)
M
M
*
TERMINAL ATTACHMENTS
IBM 1009 DATA TRANSMISSION UNIT
IBM 1013 CARD TRANSMISSION TERMINAL
IBM 7702 MAGNETIC TAPE TRANSMISSION TERMINAL
IBM 7740 COMMUNICATION CONTROL SYSTEM
IBM SYSTEM/360 WITH SIMILARLY EQUIPPED 2701'S
IBM 1050 DATA COMMUNICATION SYSTEM
EUROPEAN TELEPRINTERS (WTC ATTACHMENT)
SERIAL SYNCHRONOUS TERMINALS
BINARY SYNCHRONOUS ADAPTER
IBM 2702 TRANSMISSION CONTROL
1
15
16
31
TWO-PROCESSOR SWITCH FEATURE
31-LINE EXPANSION FEATURE
IBM 1050 DATA COMMUNICATION SYSTEM
EUROPEAN TELEPRINTERS (WT USE)

Ⓜ Multiplexor Channel

*Second channel interface feature

For more detailed configurations of the 2701, see IBM System/360 Data Communications and Acquisition Configurator, Form A22-6824.

Figure 34. System/360 Input/Output Configurator for Data Communications


2702 TRANSMISSION CONTROL
# 7935
TERMINAL CONTROL EXPANSION FEATURE (REQUIRED FOR MIXING IBM AND TELEGRAPH TERMINAL CONTROLS)
TERMINAL CONTROLS (CHOOSE AT LEAST ONE, NOT MORE THAN THREE)
LINE ADAPTERS (CHOOSE ONE PER LINE, NOT EXCEEDING 25 IBM LINE ADAPTERS TYPE 1 OR UP TO 16 IBM LINE ADAPTERS TYPE 2, OR 31 WT TELEGRAPH ADAPTERS)
#4615
IBM TERMINAL CONTROL TYPE 1
ADD. SELECTIVE SPEED FEATURE
#1065
IBM LINE ADAPTERS TYPE 1 OR 2
DATA SET LINE ADAPTER
# 3233
# 4612
# 4634
# 4635
TWO-PROCESSOR SWITCH FEATURE
REMOTE SWITCH ATTACHMENT
#2795
WT TELEGRAPH TERMINAL CONTROL
ADD. SELECTIVE SPEED FEATURE
#1065
WT TELEGRAPH LINE ADAPTER
# 2799
ADDITIONAL FEATURES
SPEED EXTENSION FEATURE (600 BPS) OPTIONAL WITH IBM TERMINAL CONTROL TYPE 1
#7387
*
OR
31-LINE EXPANSION FEATURE (200 BPS) REQUIRED WHEN MORE THAN 15 LINES ARE INSTALLED
#7955
# 3853
EXPANSION BASE REQUIRED IF IBM LINE ADAPTERS TYPE 1 ARE INSTALLED

NOTE: THE 2702 CAN BE ATTACHED TO AN IBM SYSTEM/360 MULTIPLEXOR CHANNEL

*These features are mutually exclusive.

├── NOTE: THIS SYMBOL SPECIFIES A CHANNEL CONNECTION

Figure 35. System/360 Data Communications and Acquisition Configurator for IBM 2702

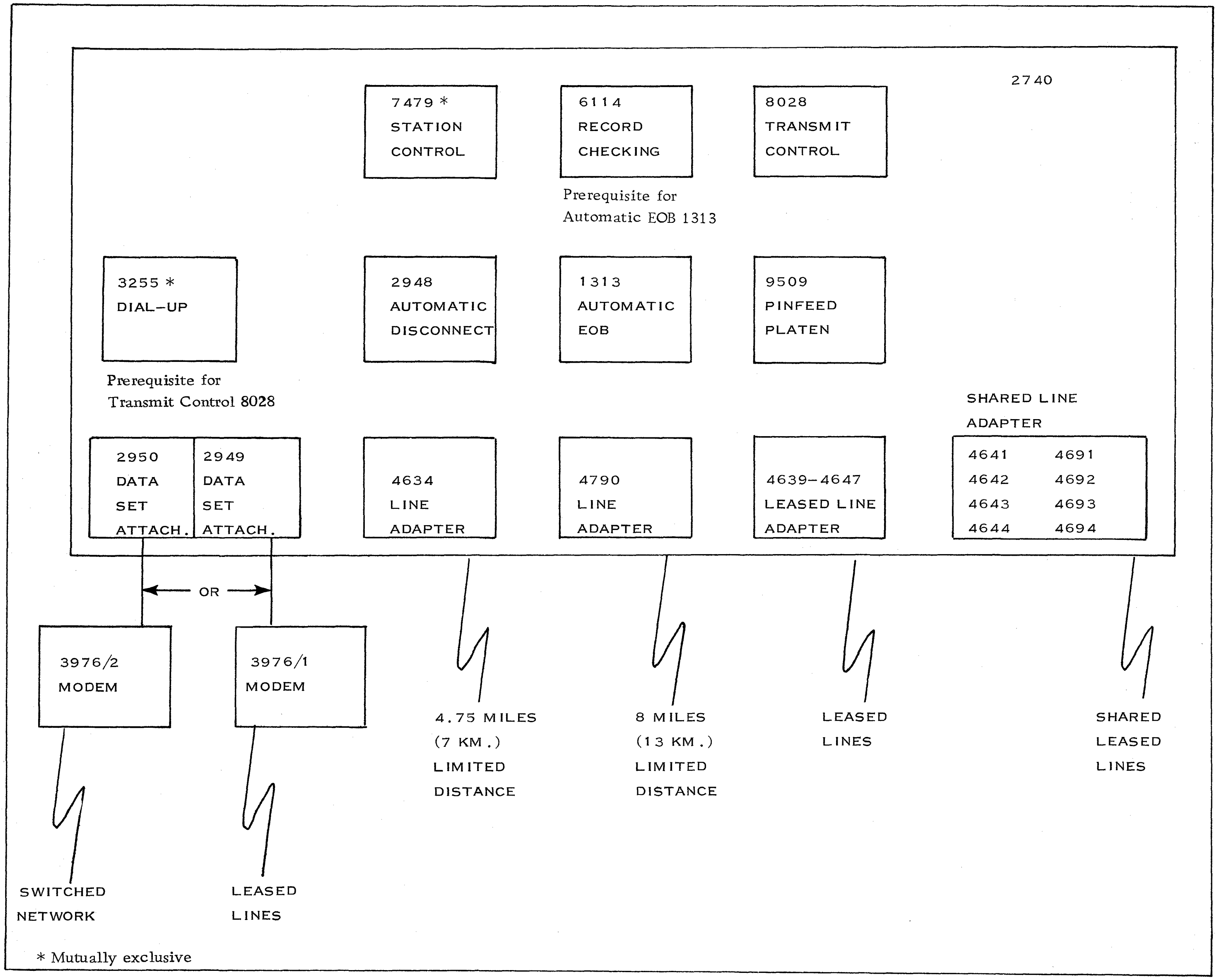


Figure 36. 2740 Communications Terminal: Configurator

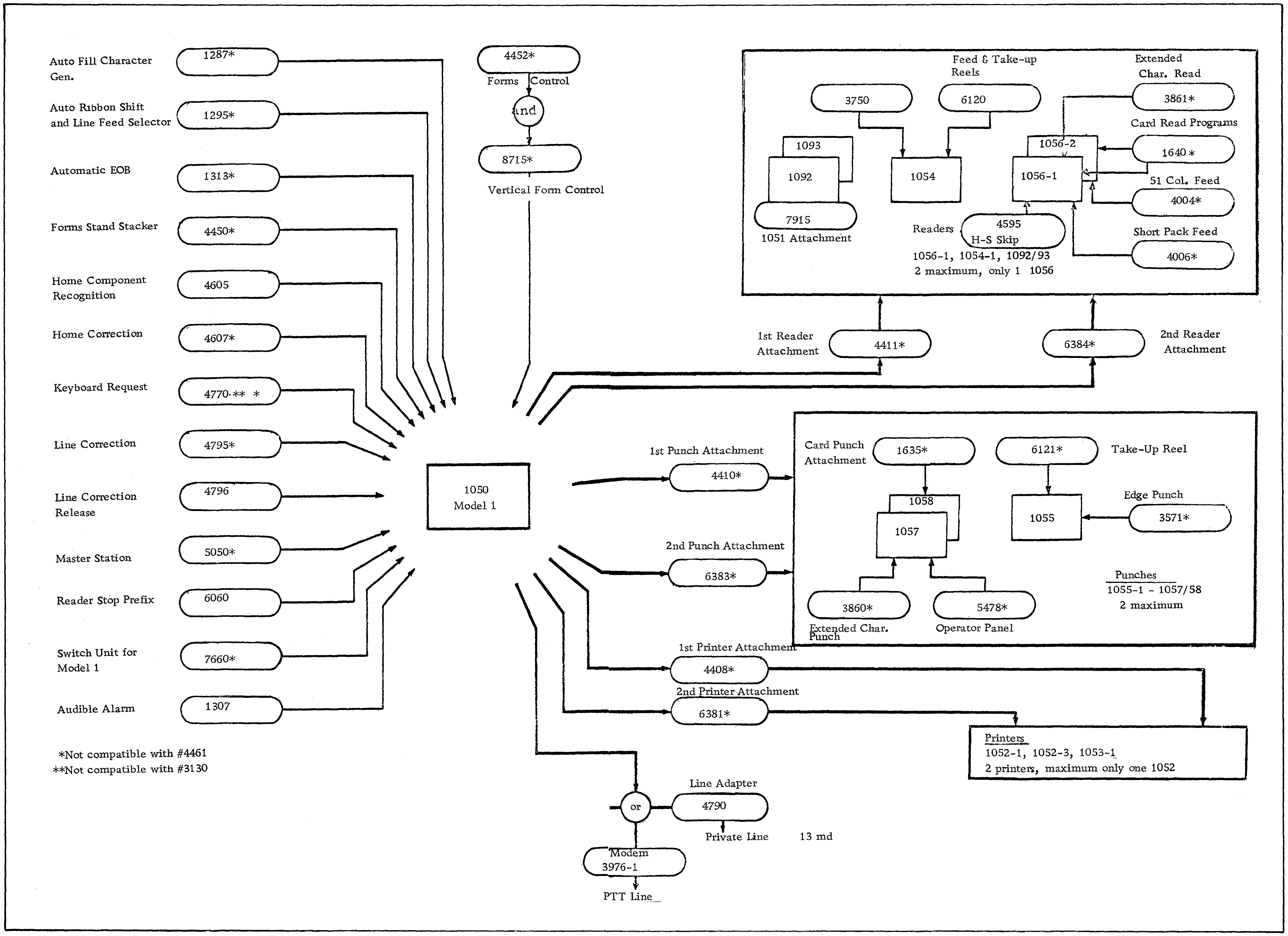


Figure 37. IBM 1050: Configurator

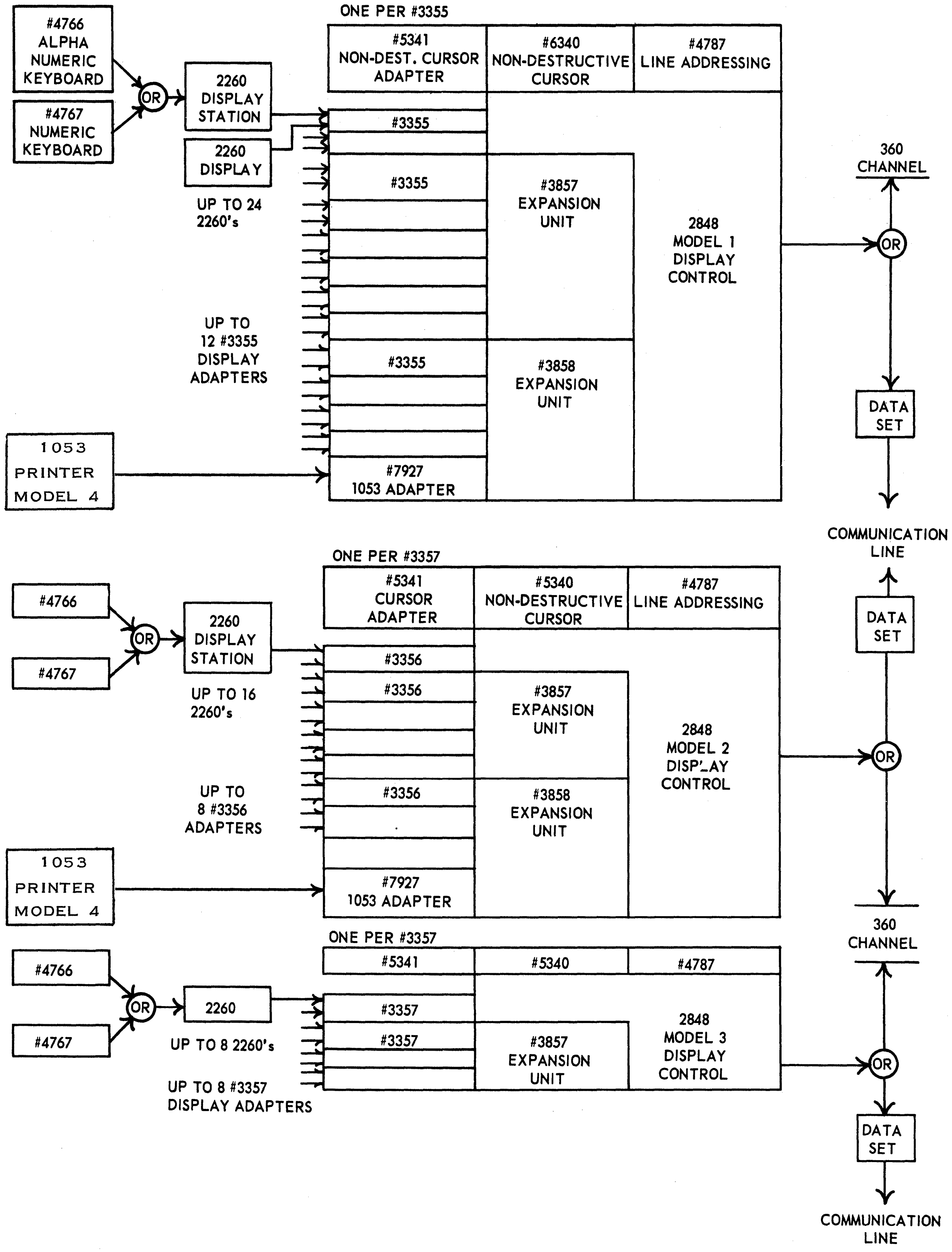
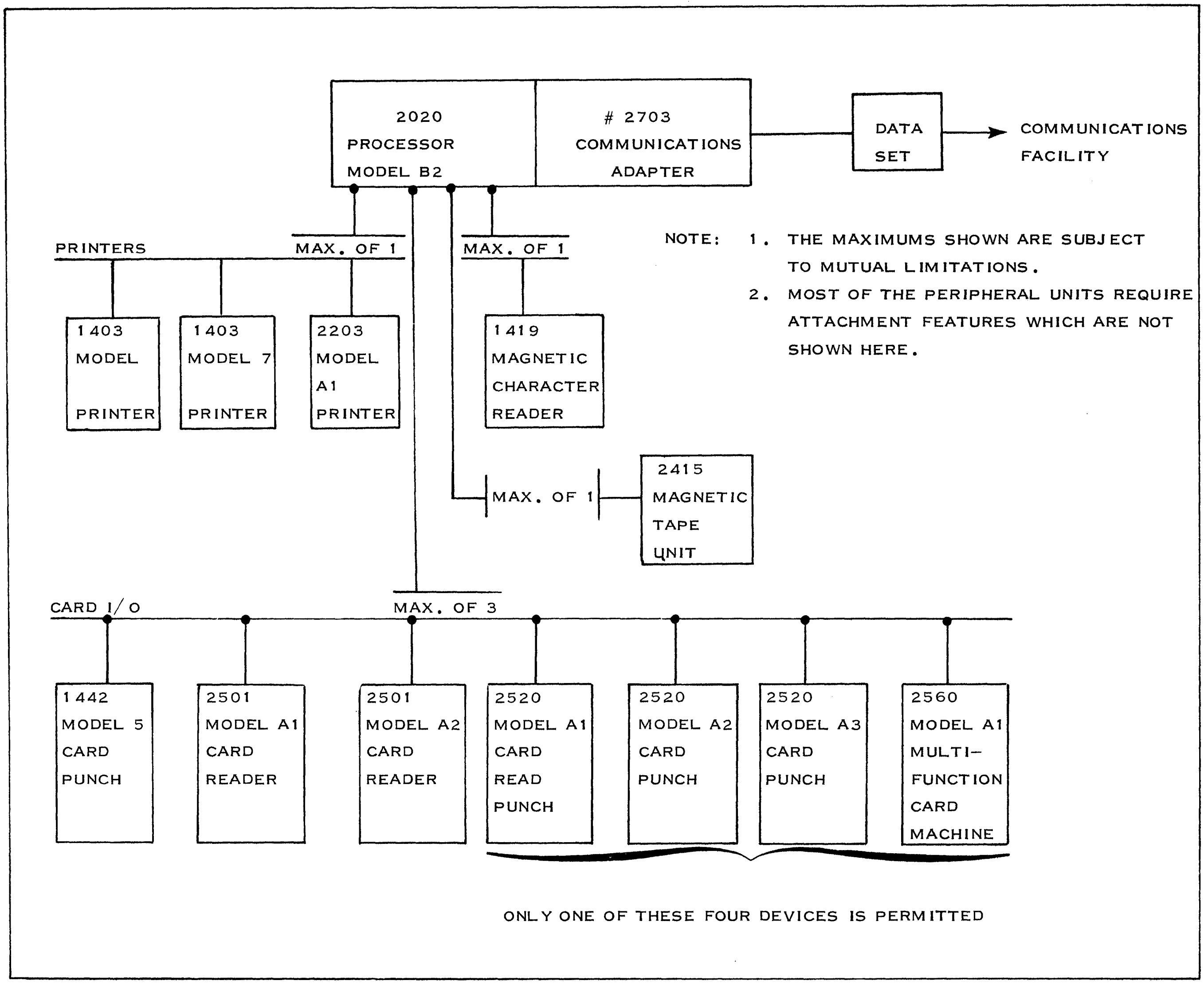


Figure 38. 2260 Display Station: Configurator



2020
PROCESSOR
MODEL B2
# 2703
COMMUNICATIONS
ADAPTER
DATA
SET
COMMUNICATIONS
FACILITY
PRINTERS
MAX. OF 1
MAX. OF 1
NOTE: 1. THE MAXIMUMS SHOWN ARE SUBJECT TO MUTUAL LIMITATIONS.
2. MOST OF THE PERIPHERAL UNITS REQUIRE ATTACHMENT FEATURES WHICH ARE NOT SHOWN HERE.
1403 MODEL PRINTER
1403 MODEL 7 PRINTER
2203 MODEL A1 PRINTER
1419 MAGNETIC CHARACTER READER
MAX. OF 1
2415 MAGNETIC TAPE UNIT
CARD I/O
MAX. OF 3
1442 MODEL 5 CARD PUNCH
2501 MODEL A1 CARD READER
2501 MODEL A2 CARD READER
2520 MODEL A1 CARD READ PUNCH
2520 MODEL A2 CARD PUNCH
2520 MODEL A3 CARD PUNCH
2560 MODEL A1 MULTI-FUNCTION CARD MACHINE
ONLY ONE OF THESE FOUR DEVICES IS PERMITTED

Figure 39. System/360, Model 20 (As a Remote Terminal): Configurator

## 3. 7 PROGRAMMING DESIGN

At the programming design level for a small CBS five major questions must be answered:

1. Can the messages be processed sequentially?

2. Are high throughput and short response time a requirement?

3. Is the message processing required to be overlapped or in asynchronous (queued) operation?

4. What programming support can be used?

5. How much core is used in total?

The answers to these questions can be found below.

### 3. 7. 1 INTERNAL OPERATION OF BTAM IN DOS AND OS

A. BTAM in DOS

The logic of BTAM contains the following routines:

1. READ/WRITE routine
2. Channel appendage routine
3. WAIT routine
4. RESETPL routine
5. REQBUF (Request Buffer) routine
6. RELBUF (Release Buffer) routine

The problem program in its relation to the synchronous operations of BTAM:

Before any message can be sent or received over a line, an OPEN macro instruction must activate the line group. That is, it:

a. organizes the buffer pool, if necessary;

b. executes, if necessary, the channel program required to initialize the telecommunication control units and data sets.

The numbers used below refer to Figure 40.

(1) The problem program requests line activity through a macro instruction: READ, WRITE or CONTROL. The macro expansion results in a linkage to the READ/WRITE routine, which creates the necessary channel program (2) and issues an execute channel program (EXCP), which results in a supervisor call (SVC) (3).

(4) The supervisor gets control, starts the line operation, and returns control to the problem program (5). The problem program's processing must be independent of the operation started in (1) and no read, write or control operation can be requested for a device on the same line. BTAM does not queue requests for line operations.

(6) At a place in the telecommunication program where further processing is dependent on the completion of one or more line operations, the problem program issues a multiple WAIT macro instruction.

(7) The WAIT routine gets control.
If awaited events have not occurred, the WAIT routine releases control to the supervisor. The supervisor gives control to a program with lower priority (9).
If the awaited user-specified number of events have occurred, the WAIT routine returns control to the problem program at the instruction following the WAIT.

(8) The problem program regains control, processes the messages received, and issues READ, WRITE or CONTROL macro instructions (1).

B. Operations on T/P-I/O Interrupts

(10) An I/O interruption gives control to the supervisor. If that interruption was initiated by a Tele-processing device, the supervisor gives control to the channel appendage routine, a routine which operates in supervisor state and is not interruptible by I/O terminations (BTAM Supervisor).

(11) The BTAM supervisor processes the interruption and returns control to the supervisor for execution of one of the following functions.

a. If the operation is not completed (negative response to polling), restart line activity and return control to the interrupted program. That program may be the Teleprocessing program, if no WAIT has been issued (multiple WAIT or BOS WAIT), or the background program, if a WAIT has been issued.

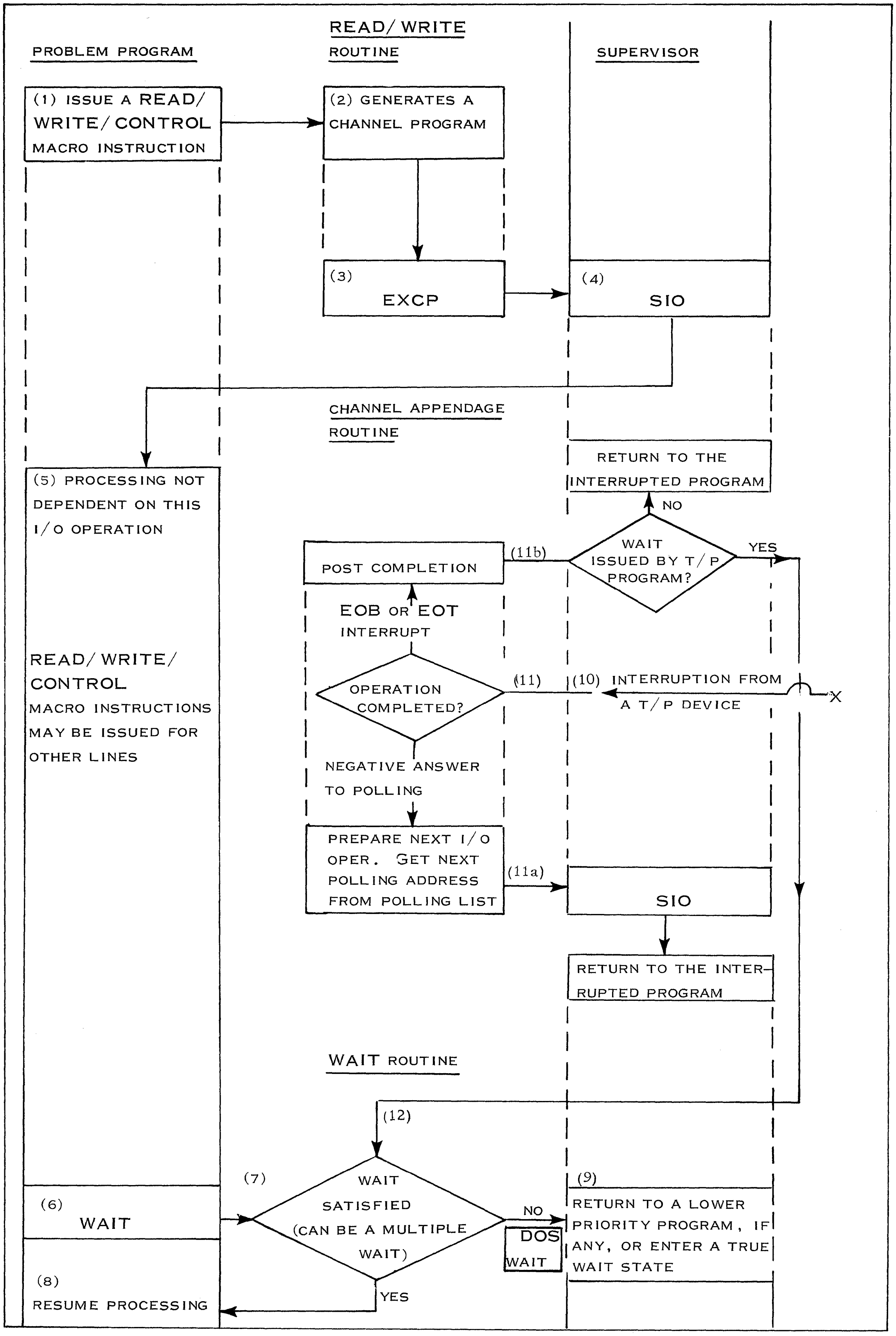

PROBLEM PROGRAM
READ/WRITE ROUTINE
SUPERVISOR
(1) ISSUE A READ/WRITE/CONTROL MACRO INSTRUCTION
(2) GENERATES A CHANNEL PROGRAM
(3) EXCP
(4) SIO
CHANNEL APPENDAGE ROUTINE
RETURN TO THE INTERRUPTED PROGRAM
(5) PROCESSING NOT DEPENDENT ON THIS I/O OPERATION
NO
POST COMPLETION
(11b)
WAIT ISSUED BY T/P PROGRAM?
YES
EOB OR EOT INTERRUPT
READ/WRITE/CONTROL MACRO INSTRUCTIONS MAY BE ISSUED FOR OTHER LINES
OPERATION COMPLETED?
(11)
(10) INTERRUPTION FROM A T/P DEVICE
X
NEGATIVE ANSWER TO POLLING
PREPARE NEXT I/O OPER. GET NEXT POLLING ADDRESS FROM POLLING LIST
(11a)
SIO
RETURN TO THE INTERRUPTED PROGRAM
WAIT ROUTINE
(12)
(7)
WAIT SATISFIED (CAN BE A MULTIPLE WAIT)
(6) WAIT
NO
DOS WAIT
(9)
RETURN TO A LOWER PRIORITY PROGRAM, IF ANY, OR ENTER A TRUE WAIT STATE
(8) RESUME PROCESSING
YES

Figure 40. DOS/BTAM General Functions

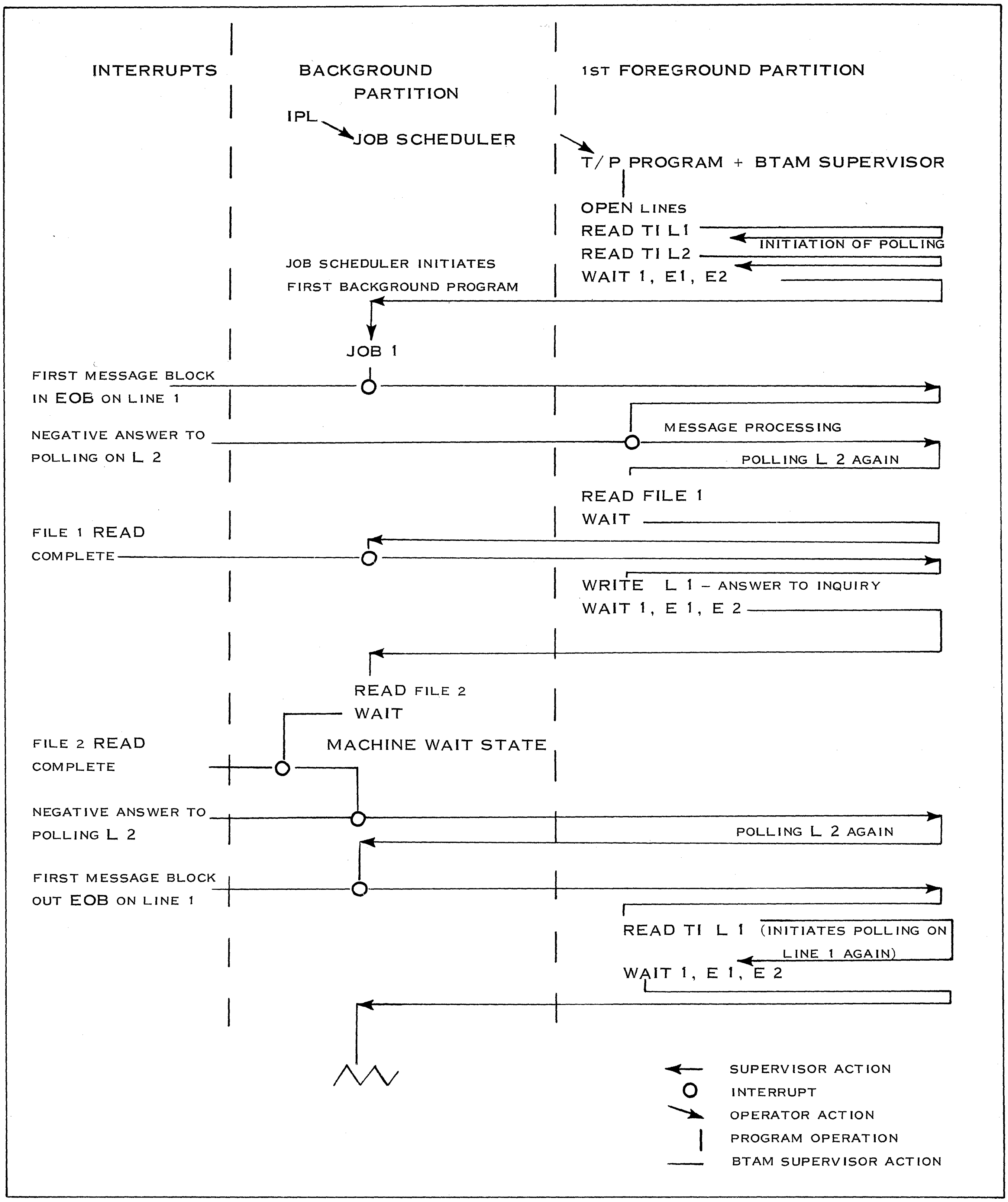


Figure 41. Control Flow for DOS (Version 2) + BTAM

b. If the operation is completed (EOB interrupt), the supervisor returns control to the interrupted program if no multiple WAIT has been issued, or to the WAIT routine if a multiple WAIT has been issued.

(12) The WAIT routine gets control. [See (7).]

Steps (1) through (9) are in sequence. Steps (10) through (12) are asynchronous with the processing of the problem program.

Control Flow for a System Using DOS and BTAM in One Partition

Figure 41 shows how control is given to the different programs after interrupts or I/O macros. For simplification, we will discuss only an inquiry application with two lines using one foreground partition. With BTAM, there is an internal supervisor in this partition. (The second foreground partition is not used in this example. If it were present, control would be given to this partition prior to the background program as long as it was not in the wait state. )

The assumed sequence of events is used merely as an example to illustrate control flow. After initial program loading and initiation of the T/P program by the operator, the T/P program starts opening the lines and initiating polling by the READ TI macro. The detailed control flow for a READ or WAIT macro is shown in Figure 41. The WAIT gives control to the partition with the next lower priority, in this case the job scheduler in the background partition.

Job 1 is started and processed until the first I/O interrupt; for example, an EOB on line 1 occurs. This means that a message has been read in and should be processed. The DOS supervisor gives control to the BTAM supervisor, which checks to see if the WAIT is satisfied (here we are waiting for one of two events, so the WAIT is satisfied) and gives control to the message-processing program. The interrupt caused by the negative response to polling on line 2 is handled by the BTAM supervisor by issuing a poll to the next terminal of line 2 (which can always be the same if only one address and a wraparound list are defined). Processing then continues until the WAIT after the READ for a file. Job 1 uses this waiting time until the WAIT is satisfied by the non-T/P-I/O interrupt.

Message-processing then continues, sending an answer back message to the terminal. The WAIT again gives control to the background program, which in turn issues a READ to a file and a WAIT. Now, no other program can use the CPU, and a true wait state is entered. Upon completion of this READ, Job 1 continues until it is interrupted by another negative response on line 2, handled by the BTAM supervisor as described above. A satisfactory reply by the terminal to the answerback message causes the next interruption of Job 1. Through both supervisors, control is given to the T/P program, which starts polling on line 1 again. If an EOB or EOT interrupt has been posted, the T/P program would start processing that message immediately. In our example, Job 1 continues.

C. BTAM in OS

As the operation of BTAM has already been described, we will only show here an overall picture of a BTAM read operation in the OS environment.

Not all BTAM macros needed for a read operation are illustrated in Figure 42. Macros such as DCB, DFTRMLST, and CLOSE are necessary and would appear somewhere in the user program. They are excluded only to simplify the diagram. A BTAM write operation could be illustrated in a similar diagram.

System Generation

This aspect of BTAM operation is shown merely as an illustration and is not intended to describe the complete system-generation procedure. It does show the necessity of describing pertinent details of the communications system hardware by use of system-generation statements that result in the forming of unit control blocks (UCB). The UCB's so created will subsequently be used by the BTAM open routine to determine the types of terminal devices involved.

Also as a result of system generation, the BTAM routines and macro definitions are included in the system library.

Assembly

All BTAM macros are represented by macro definitions in the system library. This is true for all BTAM macros, whether they are purely BTAM macros (for example, DFTRMLST, CHGNTRY) or specialized parts of system macros used by BTAM (for example, READ, WRITE).

When the Assembler language problem program using BTAM is assembled, each macro instruction contained in the program is replaced by its appropriate macro expansion. The expansion may consist of Assembler instruction statements (for example, DC, DS), symbolic machine instruction statements (executable machine instructions), or both.

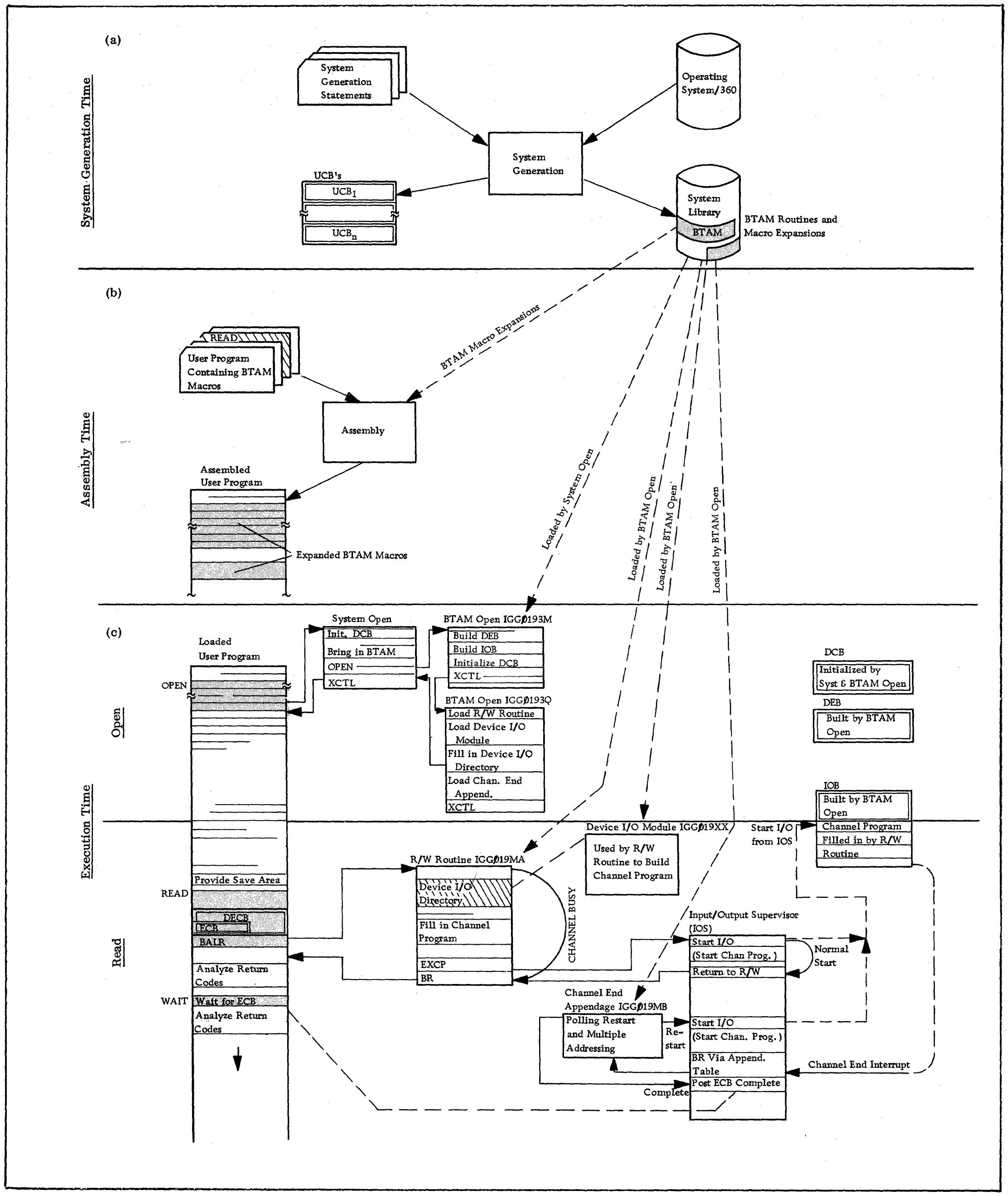


Figure 42. BTAM Read Operation

Execution

Figure 42 shows a BTAM read operation during execution of a user program and also shows those functions performed during the open and during the read operation.

Opening of the data set (the communications line group) usually occurs early in the user program. In any case, it is necessary that it be done before the data set is referenced by a READ or WRITE macro instruction. Closing of the data set (CLOSE macro) is not shown in Figure 42. This macro would normally be placed at a point in the user program where it would be executed when no more references are to be made to the data set.

Defining the data control block (DCB) with the DCB macro and the terminal list with the DFTRMLST macro is normally done in the definition section of the program (along with DC and DS statements) and, of course, does not result in the generation of executable machine instructions.

Open: Opening, or making ready for use, the communications line group data set consists of sequentially proceeding through:

1. OPEN macro expansion coding
2. System Open Routine
3. BTAM Open Routine (module 1)
4. BTAM Open Routine (module 2)

The two BTAM open modules are executed serially in the supervisor transient area. They are reentrant and operate interrupt-enable in supervisor mode.

BTAM open module 1 reserves storage for and initializes data extent blocks (DEB) and input/output control blocks (IOB). Information found in the UCB's created by system generation is used for this purpose. BTAM open module 2 is loaded and given control via an XCTL macro instruction. This module loads (via the LOAD macro) into core storage the remaining BTAM modules needed. These include the read-write routine, the channel-end appendage, and the device I/O modules for the particular terminal device involved. A directory of I/O modules is created within a section of the read-write module. To complete the open process, control is now passed from IOS to the last load module of system open and then to the user program at the point immediately following the OPEN macro expansion. The communication lines are now ready for operation.

Read: One READ macro expansion is shown in the user program in Figure 42. This represents the operaticn of a single communication line in the system. Many lines may be operated concurrently by issuing READs (or WRITEs) for each line before "waiting" for the completion of one or more.

At the point in the user program where a read operation is desired, the user must do the following in sequence:

1. Specify in register 13 the address of a save area for storing the general register values when control is passed to the read-write routine.

2. Issue a READ macro instruction.

3. Analyze the codes returned by BTAM in register 15, indicating whether the operation was initiated successfully.

4. Issue a WAIT macro instruction at the point beyond which execution is to proceed only after the read operation is complete.

5. Analyze the completion code in the ECB to determine whether the operation was completed successfully.

The expansion of a READ macro by the Assembler results in initializing a data event control block (DECB) with the parameters specified by the READ macro. The DECB is generated or updated by the macro expansion and contains within itself an ECB.

The DECB is the only direct communications link between the access method (BTAM) and the user program. The ECB within it is the entity upon which a WAIT is made (that is, a WAIT macro parameter) and in which completion is posted by the supervisor when the read operation is completed.

The prime purpose of the read-write routine is to construct the channel program for the particular type of read or write desired (that is, read initial, read repeat, etc.), making use of the device-dependent information contained in the appropriate device I/O module. Each device I/O module contains, along with special characters and a table of offsets, a number of model channel programs -- one for each type of operation. Each model channel program consists of a number of model channel command words -- there being a model CCW for every actual CCW in the channel program to be built.

Just as the DECB is considered the link between the user program and BTAM, the IOB can be considered the link between BTAM and the Input/Output Supervisor (IOS). The starting address of the channel program built by the read-write routine in the IOB field reserved for it is indicated to IOS in a field in the IOB.

Once the channel program has been built and passed tọ IOS for execution, control is returned to the user program, with return codes in register 15 to indicate whether the initiation of the read was successful. The user program may then proceed with its execution up to a point in the program where a wait is specified for completion of the read operation.

Execution of the channel program proceeds concurrently with the execution of the user program until a channel-end condition occurs, whereupon IOS regains control and enters the channel-end appendage. This condition may have been caused either by completion of the read operation or the need to update and restart the channel program, as would be the case for polling the next terminal in the terminal list after a negative polling response.

If the read operation is completed, the supervisor will post completion in the ECB specified by the READ macro. The WAIT macro referencing this ECB will then be satisfied, allowing the user program to continue its execution. At this point, the user program should analyze the completion code returned by IOS in the ECB. Further action in the user program should be based on the results of this analysis.

This completes the entire read operation, after which the user program may continue with processing or handling of the message obtained from the terminal.

## 3. 7. 2 SELECTION OF PROGRAMMING SUPPORT

This chapter should help to choose adequate programming support for a System/360 which must perform some T/P functions in addition to its main load (batch processing).

Only the DOS Version 2 -- OS Option 2 and OS Option 4 -- allows simultaneous processing of batch and T/P programs. The choice between the DOS and OS should be made after consideration of the batch processing area, for example: availability of full language compilers, need of a special access method (BPAM) or SPOOL facilities, support of needed I/O devices, and, of course, the available machine size. If there are no strong reasons for one or the other, DOS should be used because it is more efficient in time and core.

For the telecommunications access method, we have the choice between BTAM or QTAM. The main difference is that in BTAM the line control part of the T/P program is combined with the message-processing program and has to be programmed by the user. The macros provided for I/O operations and the internal supervisor of BTAM for interrupt-handling (posting EOB interrupts and reinstate polling on negative responses to polling) ease this programming. On the contrary, in QTAM, the whole line-control program, including message queuing, is separate from the message-processing program. Instead of being programmed, all the line-control functions are defined with QTAM macros, and QTAM structures the appropriate line-control program. The separation of line control and message processing in QTAM is possible only by using two tasks with different priorities. Queues are used as the interface between the line control task and the message processing task.

The following list shows some advantages and disadvantages of BTAM and QTAM.

BTAM Advantages

- uses only one partition or task
- needs less core than QTAM (for fewer functions)
- closer contact with terminal
- easier to modify

Disadvantages

- needs much programming for line control
- line control and message processing are mixed
- difficult for asynchronous operation of line control and message processing program
- all T/P processing, such as header analysis or message queuing, must be provided by the user

QTAM Advantages

- complete separation between line control and message processing
- no line control programming by user
- provides message queuing
- provides additional functions like code translation, error checks, header analysis, time stamping, logging and operator communication

Disadvantages

- needs 2 tasks
- needs more core (for more functions)

### 3. 7. 3 PROGRAMMING CONSIDERATIONS

For inquiry-type applications as well as for updating, where an answerback is required and the line is held during processing time, only one I/O area per line is necessary, since the application requires a sequential operation per line. Using BTAM in one task or partition, all messages must be processed sequentially, as waiting times for file accesses can be used only by other tasks or partitions with lower priority. One of the main advantages of QTAM, message queuing, cannot be used, since this application requires sequential operation. In a multitask environment, where different tasks are used for message processing, QTAM increases the throughput, because waiting time for file accesses of a one-message-processing task can be used for another message-processing task.

If message segmentation is used or if we have a data collection application, the lines should be polled as soon as possible after receiving a message and not only after message-processing. BTAM used in a sequential mode cannot achieve this. Either two I/O areas used in a flip-flop manner or dynamic buffering must be provided. This makes it possible to start the message-processing program with the assignment of another I/O area or buffer to the line and the initiation of polling again. The processing of one message can then be overlapped with the reception of the next one. QTAM, with its queuing facilities, provides more efficient buffering between the randomly arriving messages and the varying message-processing times. If the mean message-processing time exceeds the mean interarrival time of messages due to many file accesses, the queues will soon be filled and servicing of the lines will be delayed. This computer-bound limitation of throughput can be tolerated for short peak periods. For continuous high throughput, QTAM in an Option 4 environment should be used, as it will speed up message processing by using waiting time for file accesses for the processing of other messages.

### 3. 7. 4 CORE REQUIREMENTS

Core estimates are of essential interest in system design; not only core requirements of the programming support, but also the requirements of the application programs have to be estimated.

Supervisor

DOS II supervisor uses 8 K, including the transient area. (In fact, it is a little less, but due to the storage protection key only multiples of 2 K can be used for all partitions, including the supervisor.)

OS + Option 2 (SPS) core requirements are not yet verified.

The preliminary figures are: 20 - 25 K bytes, depending on the I/O configuration and the system-generation options selected.

Access Methods

For all non-telecommunications access methods, refer to OS Storage Estimates, Form C28-6551. Core requirements are normally between 1 and 8K bytes.

| | |
|---|---|
| The logic routines of BTAM use about | 2,700 bytes |
| or, with the dynamic buffering control | 3,100 bytes |
| Per terminal type used, add about | 100 bytes |
| Per line group, add | 48 bytes |
| Per line for the DECB, add | 36 bytes |
| Per line for DTFBT + CCB, add 80 + 16 | 96 bytes |
| Optional communication serviceability facilities | 2,500 bytes |

| | |
|---|---|
| This results in a minimum core requirement of BTAM (1 line, 1 terminal type, no translation and no dynamic buffering) of about | 3,000 bytes |
| Include communication serviceability facilities | 5,500 bytes |
| Normal size for a system using two line groups with two lines each, two different terminal types, and dynamic buffering would be about | 3,600 bytes |
| Include communication serviceability facilities | 6,100 bytes |

Note: This does not include line control and message processing programs and I/O areas or buffers. Nor does it include checkpoint and restart and console control orders or other communication serviceability features.

I/O Areas or Buffer Pools

Depending on the technique used, the number of areas or buffers used multiplied by the length gives the core buffer size. Buffer pools, which are used only in systems with many lines or widely variable message length, require additional core for their control routines (see BTAM Logic Routines, 3. 7. 1).

| TASK | TYPE | PROGRAM | K BYTES | REMARKS |
|---|---|---|---|---|
| 0 | SV | SPS (OS + OPTION 2) | 20–25 | |
| 1 | T/P | MESSAGE CONTROL PART OF T/P PROGRAM | 1–4 | AVERAGE FOR A SMALL T/P SYSTEM |
| 1 | T/P | BTAM | 3–4 | DEPENDING ON NUMBER OF LINES AND NUMBER OF DIFFERENT TERMINAL TYPES |
| 1 | T/P | BDAM, BISAM, OR BSAM | 1–8 | |
| 1 | T/P | MESSAGE–PROCESSING PART OF T/P PROGRAM | M | |
| 1 | T/P | SERVICEABILITY PROGRAM | 2.5 – 5 | ALL TOGETHER, TASK 1 IS $\geqslant$18K FOR JOB SCHEDULER |
| 2 | ANY | | $N_2$ | TASKS 2 TO 3 CAN BE USED FOR T/P UTILITY OR APPLICATION PROGRAMS |
| 3 | ANY | APPLICATION PROGRAMS | $N_3$ | N $\geq$ 18K FOR JOB SCHEDULER |
| 4 | ANY | | $N_4$ | |
| | | TOTAL CORE | $(27.5 - 46 + N_2 + N_3 + N_4 + M)$ K | |

Figure 43. Total Core Requirements for a Small CBS Using SPS

## Application Programs

There are basically three ways of estimating the core requirements of an application program:

a. If the program has been run on another machine, a conversion factor can be used.

b. If a similar application can be found, a core estimate can be based on it.

c. If neither of the above applies, the following procedure must be used:

> Based on semi-detailed flowcharts of the major programs, estimate the number of instructions; multiply the total number of instructions by the average instruction length of 4. 5 bytes; add the estimated number of bytes for tables, lists, constants, and masks; add the figures for work and overlay areas; compute the total and add 20% as a safety factor.

This procedure relates only to the largest programs which run simultaneously on the machine.

## Utilities and Other IBM Standard Programs

Depending on the application, all necessary service programs have to be chosen. Only the core requirement of those programs that have to run simultaneously with the T/P and batch programs should be taken into account. If running as a batch, only the largest one needs to be compared with the batch application programs to see whether it needs more core.

## Serviceability Programs

A description of what is required for communication serviceability facilities is given in subsection 3. 10. Most of the routines are placed on disk; therefore, an overlay area is required.

The programs that are core-resident are those that recognize a specific situation and then request the proper routine from disk; or those that must be in core at all times, such as the error statistics and checkpoint routines.

Certain masks and constants are also required for communications serviceability facilities, such as test messages for line and terminal testing and error message masks.

Tables are also required for the checkpoint and restart routines, as well as for the statistical routines.

The method used to arrive at core estimates for these programs is the same as that described for the application program.

Communications serviceability facilities will take approximately 20K bytes of disk space and a maximum of 5K bytes in core.

### Total Core Requirements for a System Using OS Option 2 (SPS) (Refer to Figure 43)

The minimum core requirement for a small T/P system with batch processing (only 2 of 4 possible tasks used), assuming a very simple message-processing program, is consequently 20 + 18 + 18 = 56K. Therefore, it can be concluded that for a batch system, with a simple T/P application within SPS, a minimum system of 64K is required.

A batch program using QISAM, the linkage editor, and the basic overlay supervisor requires, for example:

| | |
|---|---|
| Application program | 8K bytes |
| Linkage editor | 15K bytes |
| QISAM | 6K bytes |
| Overlay Supervisor | 2K bytes |
| | ≃ 31K bytes |

| | |
|---|---|
| A minimum value for a small T/P program is | ≃ 18K bytes |
| Average supervisor requirements | ≃ 22K bytes |
| Total | ≃ 71K bytes |

| PROGRAM | KEY | CORE |
|---|---|---|
| SUPERVISOR<br>+<br>TRANSIENT AREA | 0 | 8K |
| BACKGROUND | 3 | 10K (X) |
| FOREGROUND 2 | 2 | N X 2K |
| FOREGROUND 1<br>BTAM + MSG. PROC. | 1 | N X 2K |
| (X) = 10K IS THE MINIMUM CORE FOR JOB SCHEDULER<br>N = 0, 1, 2 . . . | | |

Figure 44. Core Requirements for DOS

This shows that an average batch system with a T/P application using SPS will require a 128K Model 40 system.

If more than two partitions are used, a Model 40 with at least 128K is always required.

### Total Core Requirements for a System Using DOS II

The core allocation of the different partitions is assigned during system generation. It can be altered by the operator when he initiates the foreground jobs. Core requirements are shown in Figure 44.

Using BTAM in a foreground partition requires at least 6K for a simple terminal configuration (one or two 1050 lines) with a very simple message-processing program, excluding communication serviceability facilities. The minimum core requirement is consequently 16K bytes, excluding background program. Therefore, it can be concluded that, using DOS, a small batch system with a simple T/P application requires a 32K machine.

An average system using all three partitions requires:

| | | |
|---|---|---|
| Supervisor | | 8K bytes |
| Background | | 10K bytes |
| Foreground 2 (utility) | | 2K bytes |
| Foreground 1 (T/P) | | |
| BTAM | 2. 7K bytes | |
| Line control | 2. 0K bytes | |
| Message proc. | 4. 0K bytes | |
| | 9. 0K bytes* = | 10K bytes |
| | | 30K bytes |

If the communication serviceability facilities are included, the core requirement would be 32K bytes.

| | |
|---|---|
| BTAM | 2. 7K bytes |
| Line control | 2. 0K bytes |
| Message proc. | 4. 0K bytes |
| Comm. Svc. Fac. | 2. 5K bytes |
| | 11. 2K bytes = 12K bytes |

As the core requirement is within the 32K limit in this example, a small increase in the batch program from 10 to 12K, a requirement of 4K for the foreground 2 partition or the addition of other communication serviceability facilities will increase the core requirements to 64K.

Refer to World Trade Interim Tele-processing Support (WITS) for additional programming guidance using user programs with BTAM. This book can be obtained from your T/P Coordinator or design center.

* Must be a multiple of 2K.

## 3. 8 FILE ORGANIZATION

### 3. 8. 1 KINDS OF FILES

An important part of making estimates for files is to decide what information has to be stored and how much space is required for it. In general, we can distinguish the following files:

Logical Files

These are the data sets needed for the applications, and the amount of storage required for those files depends directly on the number of records included in each data set.

Logging of Messages

This procedure is often used as a means for error, recovery, and audit procedures. It can also be used for retransmission of messages or for later processing of messages in a batch program.

Queuing

To save core storage, it may be necessary to queue the input messages, the output messages, or both, on an external device.

Programs

The different programs or parts of programs are brought into core when they are needed. There are two types of programs that are resident on file:

Application Programs

These include subroutines of the currently used application program which are not needed continuously in core storage but are called as each is needed. In addition, they normally include the complete library of tested application programs for desired processing runs.

Systems Programs

These include the total operating system needed in a given installation and all macros and subsidiary programs, such as link editor, utility programs, sorts and merges, etc.

With the exception of message logging, which can also be done on tape, for Tele-processing we need direct access devices for all the files mentioned above.

### 3. 8. 2 THREE METHODS OF FILE ORGANIZATION

Sequential

In a sequential file, the records are merely organized on the basis of their successive physical positions. This method can be used for logging the incoming and outgoing messages. This technique is used especially with magnetic tapes, where most of the records are processed each time the tape is used.

Index - Sequential

Normally, at least two index files are created. One index is the cylinder index, which can be stored in order of preference in core, or on a drum, or on the lowest average access part of a disk file. It contains a single entry for each cylinder of data file storage. This entry is composed of the highest record in each cylinder, plus the cylinder address. The second, more detailed, index is the track index. It is stored on a separate track (or tracks) within each cylinder of the data file and contains an entry for each of the data tracks within the cylinder and one for the overflow area. With this technique we need at least two seeks per record, unless the cylinder index is in core storage.

Direct Addressing

In this technique, the record key is numerically equal to the direct access storage address, or this address is derived from the record key by a simple calculation. Because the addresses of the device run in sequence, the data file is stored in sequence. This technique requires no index file, no overflow area or procedures.

The last two techniques are considered mainly for a small Tele-processing system.

When to Choose the Index-Sequential or Direct Addressing Technique

We must first decide if the direct addressing technique can be used. This makes it possible to locate any record in the file with one seek and one read. In order to use the key of a record directly as its address, a computation is required to arrive at a track or record address. For example, divide the key by the number of records per track; the quotient equals the relative track address.

The direct addressing method not only allows minimum disk time when processing at random, but is also ideal for sequential processing, since the records are written in key sequence.

A disadvantage is that there may be a large amount of unused direct access storage. A location must be reserved for every key in the file's range, even though many of them are not used. When the range of keys for a file includes a high percentage of unused keys so that direct addressing is not feasible (for example, customer numbers range from 0001 to 9999, but only 3,000 of the possible 9999 numbers are assigned), we must find a solution with the index-sequential technique.

### 3. 8. 3 DISK CAPACITY NEEDED

The external storage capacity needed depends on:

1. Number of records
2. Length of records
3. Format (blocked, unblocked)
4. Key lengths
5. Capacity per track
6. Organization method

This can be explained by an example. The file has the following characteristics:

| | |
|---|---|
| Number of records | 10,000 |
| Record length | 100 bytes (including key) |
| Key length | 10 bytes |
| Type of disk unit | 2311 |

We must consider the necessary disk capacity with unblocked records and blocked records (blocking factor 16).

Direct Addressing Technique

Unblocked Record Format:

The table, Capacity and Transmission Time Reference Card, in Form X20-1705, indicates that with 100 bytes/record (key included), 19 records of this length can be stored on each track. As we have 10,000 records, we need $\frac{10,000}{19} = 526.3$ tracks.

This means $\frac{526.3}{10} = 53$ cylinders.

Blocked Record Format:

With a blocking factor of 16, we get 1,610 byte records (one key per blocked record included).

The table indicates that two records can be stored on each track. We have $\frac{10,000}{16} = 625$ blocked records.

This means $\frac{625}{2} = 313$ tracks or $\frac{313}{10} = 32$ cylinders house the file.

Index-Sequential Technique (Figure 45)

For one track entry, we need two records of (key length + 10) bytes in the track indexes. In our example, the key is 10 characters. One track can contain 36 records of 20 positions.

For the nine prime data tracks (if we reserve one overflow track), we need only nine track entries (18 records). This means that we need half a track to store the track index. The other half can be used to store data. The position of each cylinder will be:

0.5 track-index track
8.5 prime data tracks
1 overflow track

Unblocked Record Format:

The table indicates that with records of 100 bytes we can store 19 records per track. To store 10,000 records we need $\frac{10,000}{19} = 526$ tracks.

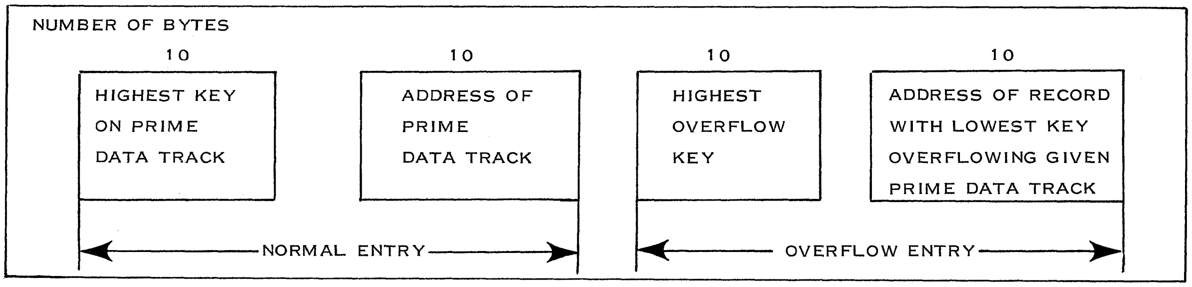

NUMBER OF BYTES
10
10
10
10
HIGHEST KEY ON PRIME DATA TRACK
ADDRESS OF PRIME DATA TRACK
HIGHEST OVERFLOW KEY
ADDRESS OF RECORD WITH LOWEST KEY OVERFLOWING GIVEN PRIME DATA TRACK
NORMAL ENTRY
OVERFLOW ENTRY

Figure 45. Index-Sequential Technique

This means (with 8.5 prime data tracks per cylinder) we need $\frac{526.5}{8.5} = 62$ cylinders.

To store the cylinder index, we need (key length + 10) bytes per cylinder entry. Therefore, we need 62 records of 20 bytes. The table indicates that with 20 bytes per record (key included), 36 records can be written on each track.

For the cylinder index we need $\frac{62}{36} = 2$ tracks.

Summarizing, we need for storing the 10,000 unblocked records 62 cylinders + 2 tracks.

Blocked Record Format:

With a blocking factor of 16, the record length will be 1,600 bytes. Ahead of this record there is a key of 10 bytes. Therefore, the length of the blocked record is 1,610 bytes (key included).

Figure 46 indicates that we can store two blocked records per track. The total number of blocked records to be stored is $\frac{10,000}{16} = 625$ records.

For these blocked records, we need $\frac{625}{2} = 313$ tracks or $\frac{313}{8.5} = 37$ cylinders (including the track indexes and overflow tracks). Figure 46 also indicates that with 20 bytes per record (key included) 36 records can be written on each track.

For the cylinder index, we need $\frac{37}{36} = 2$ tracks

Summarizing, for storing 625 blocked records (blocking factor 16), we need 37 cylinders + 2 tracks.

If we put the results from the calculations above in a table, we obtain a summary as shown in Figure 47.

Comments

Direct addressing versus index-sequential: The disk capacity calculated by using direct addressing is enough only if the keys of the 10,000 records to be stored in run sequence, with no unused key numbers. When a given key in the sequence is not used, the corresponding storage location is also not used.

The index-sequential method eliminates this problem. In the scheme above, we can see that we can use $\frac{62 - 53}{53} \times 100\% = 17\%$ more file capacity for unused key numbers. Therefore, the index-sequential method results in more efficient use of disk capacity.

Blocked versus unblocked record formats: The use of unblocked records normally lowers the gross data capacity of the disk pack. This results from the presence of inter-record gaps or unused positions. However, blocked records normally require more core storage than unblocked records because of the increased size of the data area. Additional core storage is required to hold blocking and deblocking program instructions.

### 3.8.4 DESIGN CONSIDERATIONS

The different types of disk storage devices have certain characteristics in common, and the steps taken by the system to carry through a disk operation are basically the same for all devices. These are as follows:

1. The control program places the file request in an access device queue, which may be of zero length.

| TECHNIQUE | NUMBER OF CYLINDERS | |
|---|---|---|
| | UNBLOCKED | BLOCKED |
| DIRECT ADDRESSING | 53 | 32 |
| INDEX-SEQUENTIAL | 62 | 37 |

Figure 47. Direct Addressing vs. Index-Sequential Technique

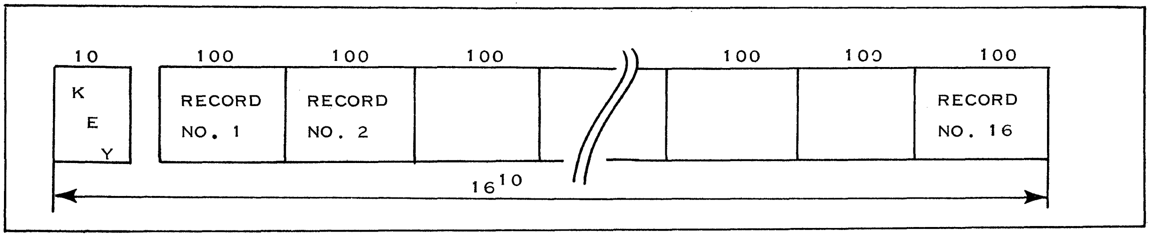


Figure 46. Blocked Record Format

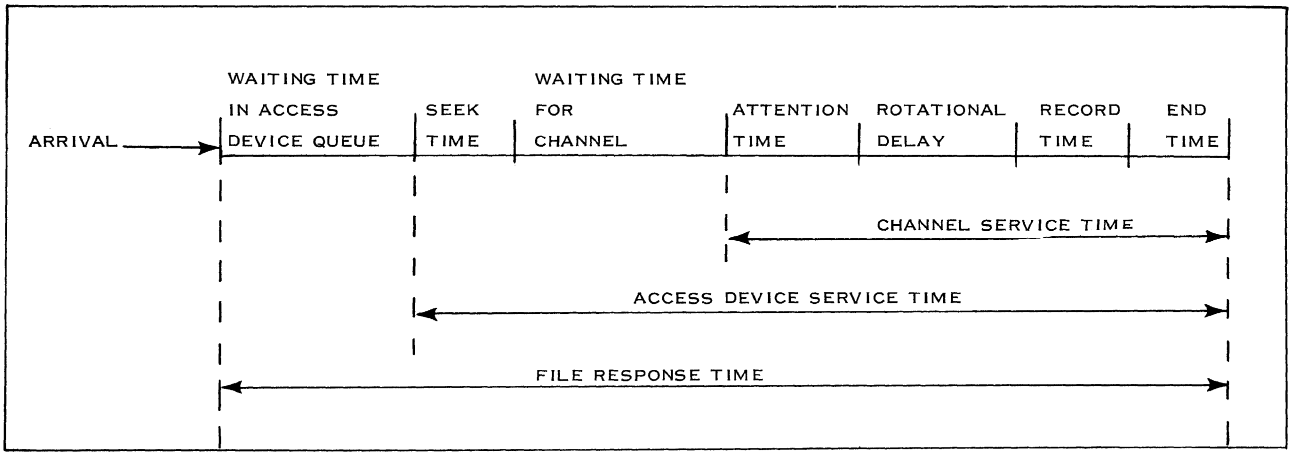


Figure 48. Timing Considerations

2. When the device and channel are available to service the request, then the CCW describing the service required is passed to the file control unit.

3. The control unit signals the arm to move to the required cylinder (seek time).

4. Then there is waiting time until the channel is free.

5. The processor is interrupted. This interrupt is serviced by the control program (attention time).

6. The rotational delay holds the channel until the record passes beneath the read/write heads.

7. The data is passed through the channel into the processor (record time).

8. When data transfer has been completed, the processor is again interrupted, the servicing of which frees the file arm and channel (end time). (See Figure 48.)

From the simplified description above, we can make rough calculations concerning some essential figures for the design of the system.

Note: Attention and end time represents the total IOCS channel control time. As we have no actual figures available and we are considering only sequential processing, we have used the read and write macro timings. For this purpose, these are conservative figures.

Utilization of Access Device

This is the proportion of total time during which a given access device is busy servicing events.

As the file response time is an exponential function (length of channel queue) of disk utilization, it is recommended that this figure should not exceed 60%.

If we have n accesses per second, the utilization of the access device will be:

$$\frac{\text{n x access device service time in milliseconds} \quad \text{x } 100\%}{1000}$$

Figure 48 shows the different components of which access device service time is composed.

We have assumed that the CPU time for handling attention time and end time is 18.4 milliseconds for System/360, Model 30, which is a rather conservative figure.

Examples:

There are 5 inquiries/second and we have one 2311 disk unit with records of 500 characters/record and direct addressing. With a character transfer rate of 156,000 characters per second, the record time will be $\frac{500}{156,000}$ = 3.2 msec.

The access device average service time for one inquiry will be:

| | |
|---|---|
| Seek time | 75 msec. |
| Attention and end time (READ macro) | 18.4 " |
| Rotational delay | 12.5 " |
| Record time | 3.2 " |
| | 109.1 " |

The access device utilization will be:

$$\frac{5 \times 109}{1000} \quad \times \quad 100\% \quad = \quad 55\% < 60\%$$

If in the above example we are concerned with an inquiry and updating application, the situation will be as follows:

We may assume that the updating of the record read from disk and the attention time and end time for a write can be done in the CPU within the rotational period of 25 minus 3. 2 msec. After this period, the record is written back in 3. 2 msec.

If there is also a write check, we must again include a rotational period of 25 ms.

The result is that the record time increases by 50 msec for inquiry and updating.

The access device service time will be 105. 7 + 50 msec = 155. 7 msec. The access device utilization is:

$$\frac{5 \times 159}{1000} \quad \times \quad 100\% \quad = \quad 80\% > 60\%$$

This is too high. The solution might be:

1. Distribute the file equally over more disk units, which decreases the number of accesses per unit.

2. Use another organization technique which results in less accesses per transaction.

3. Choose a faster type of direct access unit.

## Utilization of Channel

This is the proportion of total time during which the channel is busy servicing events. If we have n accesses per second, the utilization of the channel will be:

$$\frac{n \times \text{channel service time in msec} \times 100\%}{1000}$$

Figure 48 shows the different components of which the channel service time is composed.

### Examples

We use the same examples as above. For an inquiry application, the channel service time per access will be:

| | |
|---|---|
| Attention and end time (READ macro) | 18. 4 msec |
| Rotational delay | 12. 5 " |
| Record time | 3. 2 " |
| | 34. 1 " |

The channel utilization will be:

$$\frac{5 \times 34.1}{1000} \times 100\% = 17\%$$

The maximum recommended channel utilization is 60%. In the case concerned with an inquiry and updating application, the channel service time will be increased by 50 milliseconds, as explained previously. The channel utilization will then be:

$$\frac{5 \times 84.1}{1000} \quad \times \quad 100\% = \frac{42.5}{10} \simeq 43\%$$

In this case, if the figure were above 60%, it would be advisable to take a second selector channel and distribute the disk units over these two selector channels. Actually, if there is only one 2311 on the selector channel, the channel utilization equals the disk utilization.

## File Response Time

This is the mean elapsed time between initiation and completion of an I/O request. This includes waiting time in the access device queue, seek time, waiting time for the channel, attention time, rotational delay, record time and end time, but not the macro times.

The length of the access device queue and the waiting time for channel after the seek cannot be derived with a simple calculation. If we are within the recommended channel and disk utilization figures, a rule-of-thumb figure for the access device service time with CBS applications ($\leq$ 2 accesses/ second) is equal to 150 msec per access.

If more accurate figures are necessary for the examples calculated above, there are two possibilities:

1. We must perform mathematical calculations to establish queue lengths, service times, and the variances associated with these.

2. We can use performance curves for the disk unit involved. (See Performance Curves of IBM 2311 Disk Storage Systems, Z20-0309, for the 2311 unit. ) Examples of these curves are given in Figures 107 and 108 in subsection 4. 5.

With the help of these curves, we can also establish the 90% file response time. This means that 90% of the accesses processed by the system will have a response time less than or equal to this time. This gives a rough measure of how far above the mean the file response time is expected to vary.

## Total Response Time

File response time, as already described, is not the total time required for an I/O operation. No

IOCS time has been included for preparation of the command, checking, etc. The user interested in finding the total response time for a read or write command must include these timings.

As a rule of thumb, we can, for a System/360, Model 30, use an average of 20 msec for a READ or WRITE macro. This means that the total response time for CBS applications will be about 170 msec per access, using the average seek time of 75 msec for a 2311 for random accesses. This is a conservative number.

### 3. 8. 5 TIMINGS

A distinction is made between two classes of data-management languages. Designed for programming simplicity, the queued access languages apply to organizations with sequential properties. The programmer typically uses the GET and PUT macro instructions to retrieve and store records, and buffers are operated automatically by the system. On the other hand, the basic access languages provide for automatic device control, but not for automatic buffering and blocking. For this, the READ and WRITE macro instructions are used.

For CBS applications, we normally consider that basic languages are used because:

1. The various transactions are processed sequentially (there is no queuing - one buffer per file is enough).

2. In general, there is no need for blocking and deblocking. (Moreover, in case of logging, more messages may be logged together within one access. This can be programmed by the user himself.)

For the direct access devices, the estimated timings for file organizational methods under consideration are shown in Figure 49. Refer to the following manuals if more detailed information is required: System/360 OS Performance Factors: Reference Data, Form Y20-0001; and System/360 Performance Guidelines, Form Z20-1796.

### 3. 8. 6 OVERALL LOAD

We are considering now the overall load for the different devices (access unit, channel and CPU) due to the execution of accesses with the organization methods previously mentioned. For this we will take a Model 30. The data transfer rate of a 2311 is 156,000 bytes/second. This means $\frac{10^{-3}}{156}$ seconds/ byte/ which is about:

$$7 \times 10^{-6} \text{ seconds/byte} = 7 \times 10^{-3} \text{ msec/byte}$$

To transfer n bytes, we need: $7 \times n \times 10^{-3}$ msec.

We assume that the channel will be occupied at the maximum during the complete execution of the macro instruction. In practice, this will be considerably less. However, no exact figures are available for this.

| ACCESS | MACRO | SYSTEM/ 360 | | |
|---|---|---|---|---|
| METHOD | INSTRUCTION | MODEL 30 | MODEL 40 | MODEL 50 |
| BSAM | READ DASD | 10.3 MS | 5.3 MS | 2.7 MS |
| | WRITE DASD | 11.4 MS | 5.9 MS | 2.9 MS |
| BDAM | READ | 18.4 MS | 10.6 MS | 4.3 MS |
| | WRITE | 19.4 MS | 10.9 MS | 4.5 MS |
| BISAM | READ | 37.7 MS | 20.3 MS | 9.1 MS |
| | WRITE KN | 87.5 MS | 37.7 MS | 24.0 MS |

BSAM = BASIC SEQUENTIAL ACCESS METHOD

BDAM = BASIC DIRECT ACCESS METHOD

BISAM = BASIC INDEX-SEQUENTIAL ACCESS METHOD

Figure 49. Estimated Macro Timings for File Access Methods

| TIME ELEMENTS | 2311 | CHANNEL | CPU |
|---|---|---|---|
| | (MS) | (MS) | (MS) |
| PROCESSING TIME – READ | 10.3 | 10.3 | 10.3 |
| ROTATIONAL DELAY | 12.5 | 12.5 | |
| DATA TRANSFER | 7 X N X $10^{-3}$ | 7 X N X $10^{-3}$ | |

Figure 50. Timings for BSAM-READ for a Model 30

| TIME ELEMENTS | 2311 | CHANNEL | CPU |
|---|---|---|---|
| | (MS) | (MS) | (MS) |
| PROCESSING TIME – WRITE | 11.4 | 11.4 | 11.4 |
| ROTATIONAL DELAY | 12.5 | 12.5 | |
| DATA TRANSFER | 7 XNX $10^{-3}$ | 7X N X$10^{-3}$ | |
| WRITE CHECK | 25 | 25 | |

Figure 51. Timings for BSAM Write + Write Check for a Model 30

| TIME ELEMENTS | 2311 | CHANNEL | CPU |
|---|---|---|---|
| | (MS) | (MS) | (MS) |
| PROCESSING TIME – READ | 18.4 | 18.4 | 18.4 |
| SEEK TIME | 75 | | |
| ROTATIONAL DELAY | 12.5 | 12.5 | |
| DATA TRANSFER | 7 X N X $10^{-3}$ | 7 X N X $10^{-3}$ | |

Figure 52. Timings for BDAM - READ for a Model 30

| TIME ELEMENTS | 2311 | CHANNEL | CPU |
|---|---|---|---|
| | (MS) | (MS) | (MS) |
| PROCESSING TIME – WRITE | 19.4 | 19.4 | 19.4 |
| SEEK TIME | 75 | | |
| ROTATIONAL DELAY | 12.5 | 12.5 | |
| DATA TRANSFER | 7 XNX $10^{-3}$ | 7 X N X $10^{-3}$ | |
| WRITE CHECK | 25 | 25 | |

Figure 53. Timings for BDAM - Write + Write Check for a Model 30

| TIME ELEMENTS | 2311 | CHANNEL | CPU |
|---|---|---|---|
| | (MS) | (MS) | (MS) |
| 1ST ACCESS TO CYLINDER INDEX WITH BDAM | | | |
| PROCESSING TIME – READ | 18.4 | 18.4 | 18.4 |
| SEEK TIME | 75 | | |
| ROTATIONAL DELAY | 12.5 | 12.5 | |
| DATA TRANSFER | 7 X20X$10^{-3}$ | 7X 20X $10^{-3}$ | |
| 2ND ACCESS TO TRACK INDEX AND RECORD | | | |
| PROCESSING TIME – READ | 37.7 | 37.7 | 37.7 |
| SEEK TIME | 75 | | |
| ROTATIONAL DELAY (1 REV.) | 12.5 | 12.5 | |
| DATA TRANSFER | 7 X N X$10^{-3}$ | 7 XN X$10^{-3}$ | |

Figure 54. Timings for BISAM - Read for a Model 30

| TIME ELEMENTS | 2311 | CHANNEL | CPU |
|---|---|---|---|
| 1ST ACCESS TO CYLINDER INDEX WITH BDAM | (MS) | (MS) | (MS) |
| PROCESSING TIME READ | 18.4 | 18.4 | 18.4 |
| SEEK TIME | 75 | | |
| ROTATIONAL DELAY | 12.5 | 12.5 | |
| DATA TRANSFER | $7 \times 20 \times 10^{-3}$ | $7 \times 20 \times 10^{-3}$ | |
| 2ND ACCESS TO TRACK INDEX AND RECORD | | | |
| PROCESSING TIME – WRITE | 87.5 | 87.5 | 87.5 |
| SEEK TIME | 75 | | |
| ROTATIONAL DELAY (1 REV.) | 25 | 25 | |
| ROTATIONAL DELAY | 12.5 | 12.5 | |
| DATA TRANSFER | $7 \times N \times 10^{-3}$ | $7 \times N \times 10^{-3}$ | |
| WRITE CHECK | 25 | 25 | |

Figure 55. Timings for BISAM - Write + Write Check for a Model 30

In the examples given above, we used the BDAM access method to retrieve the cylinder index. However, it is also possible for this index to be in core or on another faster access device.

### 3. 8. 7 FILE ORGANIZATION PHILOSOPHY FOR A COMMUNICATIONS-BASED SYSTEM

Files for a CBS are designed to decrease the file response time so as to increase the throughput.

The files used and their organization for a communications application depend on the activity of the various records. For example, records for fast-moving items in a sales order entry system would be in a group of cylinders designed so that the access time is reduced and the probability of the file arm's remaining on the same cylinder or within the group of cylinders would be high.

In certain systems it is recommended that highly used logical files be duplicated either within the same physical file or on two or more separate files. Where the logical file is duplicated on one physical file, it would occupy the outermost and the innermost cylinders. The program would calculate which logical file copy is closer to where the file arm is presently located and issue the proper seek.

Major Files for a CBS

1. Queue File for Processing

This is a file where all incoming message segments are assembled into one message and placed on the proper processing queue, depending on message type.

2. Queue File for Output Message

This file queues all the output messages either by terminal, line, or group of terminals and lines.

Cylinders can be assigned to lines or terminals, based on activity.

Queuing can also be sequential. All messages can be placed on the file as they randomly occur, and the messages for individual lines can be chained together forward and backward on the file. In other words, each message would contain the file address of the message preceding it in the line or terminal queue and the file address of the message behind it. If the messages are segmented, each segment would contain the file address of the segment behind it.

A table in core indicates the file address of the first message that is designated for the individual line, line group, terminal, and/or terminal group. As a complete message is sent, the table is updated to the file address of the next message.

3. Log File

The log file can be the queue file for the output message. All that is necessary is that the messages in the queue should not be destroyed or written over after they are sent. This file then becomes the log for all output messages for the designated time period.

4. Intercept File

This is the file for queuing messages for down terminals or down lines. This file can also be incorporated in the output queue file. In some cases where the output queue is only by line, it may be necessary to have a separate location in the file for the terminal that has failed on that multi-point line.

5. Checkpoint and Restart File

This file keeps all the information necessary to recreate the system's environment before it failed.

When the checkpoint information is written on the file, it does not replace the previous checkpoint. Therefore, there are always two checkpoints kept on file. The reason for keeping the previous checkpoint is to ensure that there will always be a checkpoint even where the system fails, while the latest checkpoint is being written into the file.

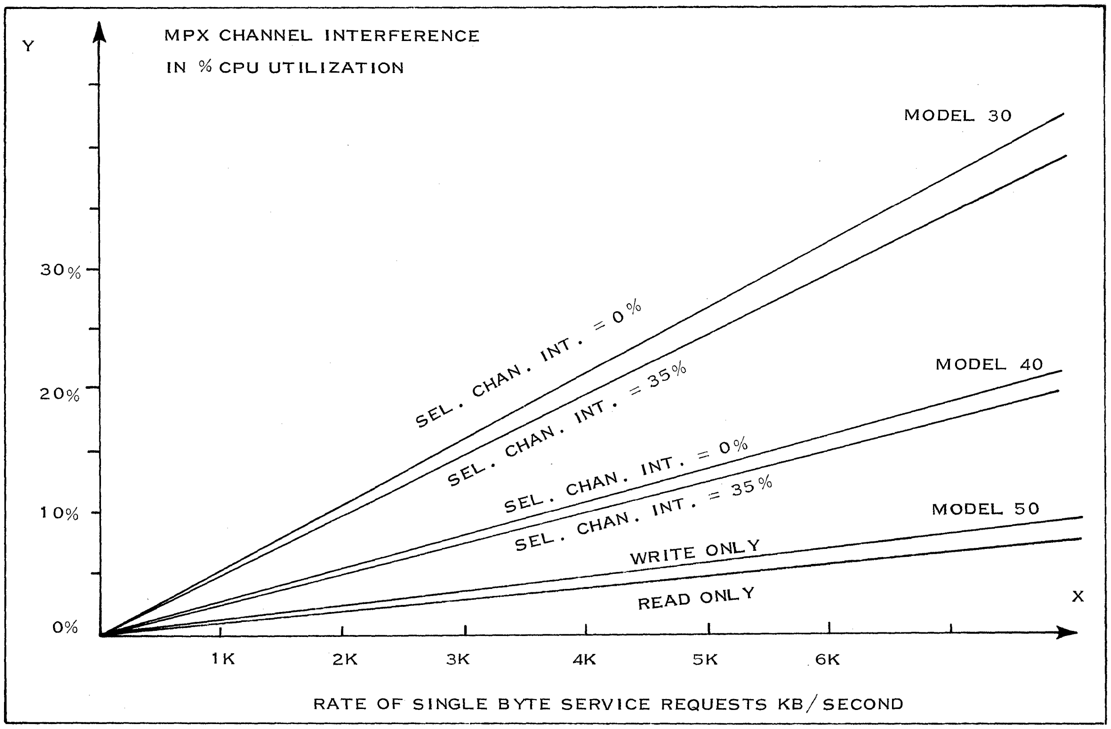


Figure 56. Data Transfer Channel Interference of a 2701 Control Unit

## 3. 9 CHANNEL LOAD AND INTERFERENCE

When studying the channel utilization of a System/360, special care must be taken concerning the possibility of overloading the data channels. Furthermore, CPU interference must be taken into account.

### 3. 9. 1 CHANNEL LOAD

An overrun condition occurs when a channel cannot accept or transfer data within time limits during a read, read backward, or write operation. This may happen only with unbuffered units or units with 1- or 2-character buffers, such as tapes, 2701, or 2702 and causes a loss of data.

An overrun causes a unit check indication in the CSW which can be sensed. This causes an interrupt to the control program. When the overrun condition involves a 2701 or 2702, the message has to be repeated.

For buffered devices, this overload causes only a loss of performance. If the I/O units are asynchronous, only the channel delay slows down the system. If the I/O units are synchronized, when the channel is ready to send or receive data it has to wait for the next machine cycle in order to be synchronized with the next I/O unit. Thus, the delay becomes more important in the case where I/O units must be synchronized.

These overload conditions are exceptional and may happen only on large systems with channels loaded at 70% or 80%. The probability of an overrun condition occurring in a small CBS is negligible.

For further information regarding exact calculation and procedure for channel load, the reader is referred to System/360/30 Channel Characteristics and Functional Evaluation, Form A24-3211; System/360/40 Functional Characteristics, Form A22-6881; and System/360/50 Functional Characteristics, Form A22-6898.

### 3. 9. 2 CPU INTERFERENCE

The interference with the CPU involves only cycle-stealing for data transfer and micro-program use by data channels.

The CPU and channels share main storage and control (micro-program) facilities. If the CPU and a channel attempt to use main storage or micro-program at the same time, the channel receives priority and the CPU stands by until both facilities become available.

Such CPU standby time (interference) reduces the CPU time available and the CPU's ability to process.

Channel interference is caused by:

1. Transfer of data to or from the main storage via a channel;

2. Execution of channel control micro-programs caused by:

   - Start I/O instructions;
   - Immediate operations (transfer of address bytes, for example);
   - Command chaining;
   - Data chaining;
   - Transfer in channel (TIC);
   - Program control interrupt;
   - Device-end/channel-end interrupts.

For exact calculations for CPU interference, the reader is referred to Average Reduction in Processing Capability Caused by Channel Interference in System/360, Models 30, 40 and 50, Form Z20-1780.

As an example, we have reproduced one graph from the referenced book (Figure 56) in order to estimate the CPU interference caused by the addition of T/P capabilities to a system. To use this graph, locate the appropriate point on the X axis (throughput) indicating characters, bytes per unit of time. Follow the vertical line up from the selected point on the X axis where it intersects with the curve for the appropriate model. This is the operating point. The amount of MPX channel interference is shown on the Y axis directly to the left of the operating point.

For example, a system driving six 1050 lines has a maximum data rate of 90 characters/second. The resulting interference due to six 1050 lines is very low. In fact, it does not reach 1% for a Model 30 System. Therefore, the CPU interference due to the addition of a small T/P capability fo the MPX channel is negligible and has little effect on the overall performance of the system.

## 3. 10 PROGRAMMING ERROR CONTROL AND RECOVERY (Communications Serviceability Facilities)

Error control and recovery techniques are of prime importance to a CBS in its design and installation. Even though this Handbook considers problems relating only to a small CBS, error control and recovery techniques are necessary for message and file protection. These techniques aid in defining file organization, core size, configurations, and operational procedures, and they have a direct bearing on the backup, and restart and recovery procedures.

For a communications-based system, error control and recovery techniques consist of the following:

1. Error detection and recovery procedures within the programming systems, known as "communications serviceability facilities". This includes error detection within the hardware and the operation of the programming systems with this hardware capability.

2. Message and file protection.

### 3. 10. 1 COMMUNICATIONS SERVICEABILITY FACILITIES.

Every T/P system requires error detection of the hardware, lines, data sets, and terminals in order to enable the programming systems or the operators to take corrective action. These facilities also enable the system's online application program to perform message and file protection. They can also be considered online diagnostics.

Some of these facilities are already available in existing Type II communication control programs for the System/360 and will be planned for future versions of Type I programs. At this time, these facilities must be provided by the design and installation team.

Communications serviceability facilities should provide programs to perform the following functions (these functions are common to all CBS's, whether they be 1050, 2740, 2260, or STR systems):

1. Checkpoint and Restart

   This gives the user the capability of checkpointing certain information (pointers, counters, sequence numbers, etc. ) that he deems necessary at specified time intervals and/or at the initiation or completion of hardware or programming operations. He is also able to perform a restart operation with the information that was previously checkpointed and recreate the environment of the system prior to failure.

2. Error Statistics (Counts)

   These programs are for accumulating the number of errors by type for lines and terminals and are used by the C. E. during P. M. time. This information is placed on the disk file to be kept for future reference by the C. E. or the user.

3. Error Recovery Procedures

   These programs define errors, isolate error sources, such as the transmission control unit, line, data set or terminals, and, finally, call in the proper diagnostic programs to test the hardware in an online environment.

4. Line Test

   Standard error recovery procedures cause retransmission of messages which are affected by transmission errors on a line. Should retransmission (n times) fail to result in an acceptable message, diagnostic service routines are provided for one or more of the following actions:

   - Place the faulty line in test status by sending the stored test message over the line. The test continues until a predetermined number of test messages is sent without recurrence of the error condition.

   - Send a message to the operator to inform him of the action that has been taken.

5. Online Terminal Test

   This procedure would be activated by a special coded input message from the terminal to be tested or the central operator (terminal) console. It can be used by the operator or C. E. to request that certain terminal tests be performed.

6. Isolation of errors to a control unit or communications line through the use of the

Autowrap (2702) or Diagnostic Read/Write (2701) facility:

In many cases the error conditions can be isolated to the control unit or communications line. There are various methods that can be used for error isolation to determine whether the error is due to the terminal, data set, communication line, or the transmission control unit (TCU):

- Isolation of Error Between TCU and Data Set: The 2701 and 2702, when interfacing a data set, can detect errors at the interfaces. The CPU program can then test the front end of the TCU and determine whether the error is caused by the data set or the TCU.

- Isolation of Error Between TCU and Communication Line: The TCU hardware and the computer program working together are able to recognize whether or not a terminal is responding correctly to addressing or polling, if a message is not completed, or if there are errors in sending a message to a terminal.

In the case of a point-to-point line, the only error isolation possible that can be performed by the TCU and CPU programs is the isolation between the TCU and the communication line-data set-terminal combination. It is then up to the C. E. and the PTT service men to investigate the error further. The C. E. will be able to use the CPU program to issue test messages so that the line and terminal can be tested and the error or failure isolated.

In the case of a multi-point configuration, the TCU and CPU programs can isolate errors and failures caused by the TCU, the communication line or link, or the terminal-data set combination.

For example, where a terminal does not answer a poll, the following steps could be initiated (this example can apply to the 1050, 2740 or 2260):

i. Poll the same terminal again for a predetermined number of retries.

ii. If the terminal still does not answer, place the terminal in a down state by setting a flag in the terminal status table.

iii. Notify the supervisor through his terminal that the particular terminal may be in a failed state.

iv. Poll the other terminals on the same line and follow the same procedures indicated in i, ii, and iii.

v. If the other terminals on the same communication link do not answer, it can be concluded that the fault lies either in the line or the TCU.

vi. The program then can initiate its online diagnostics to test the front end of the TCU to determine whether the error is caused by the TCU or the line.

vii. The program will then notify the supervisor through his terminal of the cause of the error or failure.

viii. If all other terminals on the same communication link answer when polled, then the first (terminal data set) combination is considered to be in a failed state. It is not possible for the central system (TCU and CPU) to isolate errors between a terminal and its associated data set.

ix. The supervisor will then call the proper people at the remote site and inform them that their terminal is not operating properly and that they must contact their local IBM customer engineer and PTT serviceman to determine whether the error or failure is caused by the data set or terminal.

In the design phase, error recovery procedures must be considered in the light of customer and IBM requirements, in order that estimates of core utilization calculations and disk space utilization include the additions caused by the error recovery procedures.

7. Operator Awareness

This factor is required in the operation of a communication (T/P) system. The central operator becomes aware of the condition of the system through terminal console printouts which are initiated through error recovery programs and diagnostics.

8. Console Control Orders

These will be input messages from the central operator's control console to initiate programs to perform the following types of functions:

- Activate or deactivate a line
- Activate or deactivate a terminal
- Place a terminal in test status so that the online terminal test facilities can be used
- Request that Autowrap or diagnostic read/write routines be run to ensure that the control unit is operating properly
- Request traffic statistics by terminal or line
- Request error statistics
- Alter core
- Display core
- Initiate restart routine
- Initiate retransmission of messages to other terminals
- Reassign subchannels and devices

To design the communication serviceability facilities, the systems engineer must be familiar with 2701 and 2702 hardware error detection capabilities, such as:

- No response to a poll or a selection of a terminal
- Detection of an open line
- Detection of errors caused between the TCU and data set
- Echo-checking -- ascertains that, in fact, a proper bit was sent down the line
- Stop bit error detection -- recognizes that after receiving a start bit and the required number of data bits, a stop bit has not been received.
- LRC checking and parity checking

In most cases, a terminal is required at the central system location for operator use. This is referred to as the supervisor (terminal) console. This terminal is used for operator awareness and console control orders.

When operating in a DOS environment, it is possible to use the actual 1052 console for the Models 50 and 40, and the 1051-1052 console for the Model 30 as the supervisor console to operate with the Tele-processing program.

When operating within an SPS environment, it is not possible to use the System/360 console for the Tele-processing supervisor terminal because the Master Scheduler module of the SPS recognizes only certain orders from the console and rejects others, mainly those concerned with interfacing the T/P program. The Master Scheduler does not allow the console to have direct communication with the Tele-processing program. Therefore, it is nearly always necessary to propose an additional terminal at the central location to provide the supervisor terminal function, or to designate a terminal that will be at the central site to have the supervisor function in addition to its normal functions.

DOS/BTAM has the following communications serviceability facilities:

- Error recovery procedures (optional)
- Error counts (optional)
- Online terminal test facilities (optional)

### 3. 10. 2 MESSAGE AND FILE PROTECTION

The methods described here can be used for all three CB systems discussed in this Handbook.

The message and file protection methods vary with the application and the customer's requirements. For a small CBS, a standard method for message and file protection can be devised, since throughput and response requirements are not strict. Message protection can be achieved as follows:

1. Inquiry System

   For every input to the system from the terminal there is an output. Therefore, if the terminal receives an incomplete answerback or no answerback at the time of a failure of the central, modem, line, or terminal, it knows that at restart time it must reenter its last incomplete message. In this case, no messages can get lost.

2. Inquiry and Updating System

   An internal message serial number (system serial number) is assigned to each input message and this number is carried by the message and the associated output message produced by it, until the output message is completely transmitted to the terminal. For message protection within this type of system, the same procedure is followed as in the inquiry system, except that the answer is not sent to the inquiring terminals until the proper files have been updated.

3. Data Collection, Data Distribution, and Message Switching Systems

A terminal status table is located in core. This table contains the last input message sequence number from a terminal and the last output message sequence number to the same terminal.

The terminals attach a sequence number to all input messages.

The CPU checks the sequence numbers to make sure that, in fact, the messages from each terminal are in sequence. If a message is out of sequence, the CPU notifies the terminal and the central operator. This situation indicates whether a message has been lost or the terminal operator is in error.

All output messages are sequentially numbered for each terminal. This number is transmitted with the message.

If required, before the output message is transmitted, it is logged on disk or tape.

A checkpoint is taken at specified time intervals and the terminal status tables with the input and output sequence numbers, system sequence number, the line status tables, file addresses, statistics, etc., are written on disk.

There are always two checkpoints on disk to protect against a failure when writing the checkpoint. At restart time, the checkpoint information is used to update the terminal status tables with the input and output sequence numbers and the file address pointers to indicate where the next messages will be logged and the address of the next message in a line queue is to be transmitted.

The restart program sends a message to the terminal based on the last input sequence number and tells the terminal that the last message received had this sequence number. The terminal then sends all messages that have numbers higher than the indicated sequence number.

On transmitting to a terminal, the first message transmitted to the terminal is based on the output sequence number and system sequence number. The message is sent to the terminal with an indication that it may be a duplicate.

The checkpoint period usually depends on the average transmission time of the message. If the average message takes 200 character-times on a ten-character-per-second line, a checkpoint may be taken every 20 seconds to protect against message loss. If the system fails between checkpoints, any message being sent or received would be interrupted.

File protection can be achieved as follows:

1. Inquiry System

This system does not require any file protection, since it does not add, delete, or modify a file record.

2. Inquiry and Updating System

In this system, file protection is necessary. The use of the system sequence number is employed at this time.

Each disk track is appended with the system sequence number of the last message altering it.

The input messages are logged and queued for processing with their associated system sequence number.

Prior to sending a message, the output message is logged and queued with its associated system sequence number.

The checkpoint file indicates at restart time the last system sequence number with which a completed message was associated.

The restart program uses the system sequence number indicated on the track to prevent the updating of a record by the same message, if failure occurred during message processing.

It is noted here that the system designer must remember to include the disk accesses required per transaction for message and file protection in his CPU, channel, and disk utilization calculations.

### 3.10.3 1050 ERROR RECOVERY PROCEDURES

This subsection contains detailed descriptions of some of the 1050 ERP's when operating with a 2701 or 2702. Its purpose is to show the S. E. the level of detail that is required at installation time.

1. General Considerations

When certain errors occur, there are a number of retries that must be performed by the user, such as polling retries, when there is no response from a terminal.

Statistical information is collected and stored by the program, to be printed out for the customer or C. E. on demand, such as the number of data checks that have occurred per line and per terminal when sending and when receiving.

2. I/O Errors and Interrogation of Channel Status Word

An I/O error causes an interrupt condition. The channel status word should be interrogated and action taken. Upon initial selection, certain interrupts should not occur, such as attention, status modifier, and control unit end. If these do occur, the program should notify the supervisor. All other errors, such as incorrect length, program check, interface control check, etc. , must also be indicated to the supervisor, with the exception of the unit check interruption and unit exception. In the case of unit check, the program must analyze the sense byte.

Each supervisor message must contain the following minimum information:

a. A message code

b. Channel, control unit, line number and terminal

c. Type of error

3. Sense Byte Interrogation by the CPU Program

A sense operation must be performed whenever the unit check bit is present in the CSW. The sense byte further defines the conditions causing the unit check.

At initial selection, all errors that occur, except the bus-out check, must be indicated to the supervisor by an error message. The bus-out check requires up to three retries before notifying the supervisor.

The sense byte indicates the following conditions:

- Command reject
- Intervention required
- Bus-out check
- Equipment check
- Data check
- Overrun
- Receiving
- Time-out

4. Equipment Check

Equipment check has first priority. When a read or write command is issued, this error should not occur. If it does, the supervisor should be notified after a number of retries.

5. Bus-out Check

Bus-out check has second priority. This error should not occur with a read operation. If it does, the supervisor should be notified. This error can occur with a write operation indicating wrong parity. After a number of retries, the supervisor must be notified.

6. Command Reject

Command Reject has third priority. This error should not occur. If it does, the supervisor must be notified.

7. Overrun

Overrun has fourth priority. This error should not occur during a write operation. This error is caused when character service is not honored by the channel within the character interval of the communication line. Therefore, it can occur only during a read. When this error occurs, the customer's program is notified and a retransmit message request is usually sent to the terminal.

8. Intervention Required

Intervention required has fifth priority. If this error occurs during a write, break, or prepare command, the power is off in the attached data set or in some other non-operational mode. The other error which might have occurred would be a program error that failed to issue and execute the enable command.

When this error occurs on a 1050 line, the following steps should be taken:

- Issue the enable command and repeat the operation.
- If the intervention required occurs again, issue a diagnostic write command for the 2701, or the Autowrap command for the 2702.

- If the intervention required occurs again, the 2701 or 2702 is in a failed condition and this must be indicated to the supervisor.

- If the intervention required does not occur, send a message to the supervisor indicating the data set that has failed.

  If this error occurs during a read, inhibit, or search operation, it means that:

  - Power is off in the attached data set or is not in the proper operational mode;

  - Failure to execute the enable command;

  - Continuous space signal received for more than one-character time.

### 3. 10. 4 CONCLUSION

This subsection of the Handbook has supplied the user with the basic information necessary to arrive at the type of error control and recovery routines required for his system and to enable him to make valid core estimates and proper configurations. It has also indicated to the reader the sort of detailed knowledge that is required at installation time to implement the error control and recovery programs.

## 3. 11 BACKUP AND RESTART AND RECOVERY PROCEDURES

Backup and restart and recovery procedures should be considered in the system's design, but are developed in detail at system implementation time. The procedures described in this section do not discriminate between types of CBS's as to whether they are 1050, 2740, 2260, or STR. The user can apply these procedures to the system he is interested in, with minor modifications where necessary.

For a communications-based system, backup and restart and recovery procedures consist of some or all of the following:

1. Error and failure detection by the terminal and terminal operator

2. Backup procedures at the central system in case one of the main components fails

3. Manual backup procedures at the terminal locations if the terminal fails

4. Restart and recovery procedures at the central after the failed unit(s) has been repaired

5. Restart and recovery procedures at the terminal location after the line's modem or terminal has been repaired

6. An operating manual describing the steps to be taken for backup, restart and recovery for various conditions

### 3. 11. 1 ERROR AND FAILURE DETECTION BY THE TERMINAL AND TERMINAL OPERATOR

The terminal and terminal operator must be able to recognize errors and failure caused by the terminal, line, or CPU. Recognition of errors and failures is based upon specified procedures.

The computer is able to detect the failure of a terminal and data set combination and notify the remote site through its communication serviceability facilities. The terminals detect if the computer center or line (line and/or data set) has failed by using the following recommended procedures:

A. In the event of a failure at the computer center or of a communication line at the same time the terminals are placed on line, at the beginning of the day, the following should occur:

1. The terminals would know that the center or line is down when they do not receive an automatic answerback to their good-morning message.

2. The terminal operator then notifies the supervisor that the center or line is down.

3. The supervisor phones or telexes the computer center to pinpoint whether the failure is at the computer center or a communication line.

   An online failure of a communication line or the computer center could be caused by other line errors. The computer center must be capable of recovering from all errors that occur during any periods in which excessive error rates are not present. The method of recovery from errors can be retransmissions, error check and correction, and insignificance of error in the system application.

B. When a failure occurs in the online system during the day, the terminals will be in various phases of the input response cycle. The various terminal situations and recommended procedures follow:

1. A situation where the operator is in the process of entering a message: The first procedure is to place the terminal in the offline mode. For example, if this is an order processing system, using 1050's (as in the <u>1050 Example</u>, subsection 4. 5), orders can be processed on the keyboard printer while it is offline to produce a typed shipping order and picker tag. Other messages can be typed, in the sequence in which they appear for a record of messages that have to be sent to the computer.

   The next procedure begins when the computer center is available to the terminal. First, place the terminal in the online mode. Next, enter all high-priority messages first. These include the high-priority orders that have already been processed offline. The processed orders will be used to update the inventory at the computer center.

2. A situation where the terminal operator is in the process of receiving a response and at least one character has been printed at the terminal: Follow the same procedures mentioned above, except in the case of reentering all the messages. If the first message was of a type to update the inventory, it should not be entered, because the inventory has already been updated by it and only the response to this message was interrupted.

3. A situation where the terminal operator enters the entire message, but did not receive a response: Follow the same procedures described in step 1 above. The next procedure begins when the computer center is available to the terminal. First, place the terminal in the online mode. Message types other than those that affect the customer or item files will be reentered along with their transaction numbers. For message types that do have an effect, the files and inquiry message are entered first to determine whether the incomplete message has resulted in the updating of one of the files. This is performed at the CPU and has been described in File and Message Protection. If the message did not affect the files, it will be reentered. All succeeding offline messages must also be reentered in order of priority.

### 3. 11. 2 BACKUP PROCEDURES AT THE CENTRAL SYSTEM WHEN ONE OF THE MAIN COMPONENTS FAILS

If one of the customer's requirements is to continue the T/P application throughout its specified time, the central system must have backup procedures in case of failure of one of its major units.

When a customer does require full-time operation of his T/P application, a method must be devised to enable him to continue operating during the time the failed units are being repaired. The backup method to continue his operation can be manual operation or a second system. Since, by definition, the small T/P system is simplexed, as far as the CPU is concerned there can be no standby system backup requiring programmed switchover.

At the central site the components that are necessary to perform the T/P application are: terminal control unit, CPU, file control unit, file, and console terminal. If one of these units is not available, the central system cannot perform its T/P function. It is also understood here that at least one remote terminal and its associated communications facility is a necessary requirement to perform the T/P function.

Example of Backup Methods

To illustrate the method of backup for the individual at the central system, an example is shown based on the configuration in Figure 57:

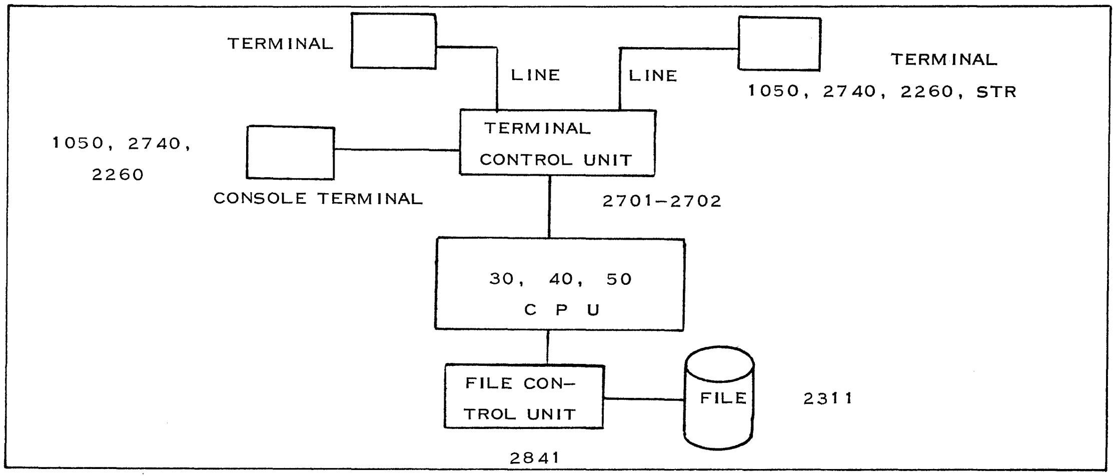


Figure 57. Configuration Illustrating Backup Method

A. Failure of the Console Terminal:

1. If possible, one of the remote terminals will be used as an alternative console.

2. If the system has a card reader and printer attached (which is not shown in Figure 57), this combination may be used as an alternative console.

3. If the customer is not concerned with operator awareness of errors or the entering of console control orders during the period the console is being repaired, it is not necessary to consider an alternative console. In that event, a failure of the console does not affect the availability of the central system for the T/P application.

B. Failure of the File:

1. In almost all cases, failure of the file is due to the drive. If the file is of the 2311 type and there are other drives available, the module located on the failed drive can be placed on the available drive. The operator can then notify the CPU through the console that the logical file has a new subchannel assignment.

2. If there are no other drives available, the system is considered unavailable for T/P, and manual backup procedures must be initiated. Manual backup procedures could consist of (a) notifying all the terminal operators by telephone that the central system has failed. (b) The terminal operators would then request information from the center via telephone. (c) The center has access to summary information in hard copy, which would be used to answer the terminal inquiries.

3. If there is no information available at the center, the system becomes totally unavailable and backup for the center may take place at the terminal location. For example, if the terminal is located in a warehouse and the online operation is for a sales order entry, the order can be processed manually at the terminal location and a picking tag and shipping order can be prepared in a home-loop operation of the terminal.

4. If there is an inquiry application, periodic printouts at the terminal locations, by the central, of pertinent information can be used to answer most inquiries in case the central system fails. For a sales order entry system where the central system keeps the inventory of all the remote warehouses, a list of all the items unavailable at the individual warehouses is periodically printed out at the terminal, since most of the inquiries are concerned with the availability of items.

5. An additional backup method is the use of a master terminal to poll and select the remote terminals. For this type of operation, a patch board is required for the communication lines. The patch board is a panel which allows the connection and disconnection of lines to the transmission control unit and also enables the customer to attach (patch) a master terminal to any of the lines to perform polling, selecting, monitoring, and testing. The master terminal could be used to allow the terminals to send their inquiries to the central office, where the information is available.

C. Failure of the File Control Unit:

1. If there is only one file control unit in the system, the central system becomes unavailable and the same procedures can be followed as in Failure of the File, paragraphs B. 2 to B. 5 above.

2. If there is an additional file control unit and available drive, the same procedure can be followed as in paragraph B. 1 above.

D. Failure of the Terminal Control Unit:

1. The central supervisor can telephone the terminals and notify them of the failure. The terminals can then request inquiries from the central office over the phone. The supervisor and operators at the center then send

inquiries to the CPU through a card reader and get the answerback on the printer.

2. If a card reader and printer do not exist and the console terminal is controlled by the terminal control unit, the procedures to follow would be as described in paragraphs B. 2 to B. 5 above.

E. Failure of the CPU

The procedures to be followed in this case are as described in paragraphs B. 2 to B. 5 above.

### 3. 11. 3 MANUAL BACKUP PROCEDURES AT THE TERMINAL LOCATIONS

In the case of a failed terminal, there are a number of methods that can be used to back up the T/P operation. These procedures can apply directly to the examples given in this Handbook.

A. The central system can recognize the failure of the terminal and/or the communication line.

1. The central system then notifies the central operator that a terminal or line is in failed status through the use of the communication serviceability facilities.

2. The operator notifies the supervisor and the supervisor calls the remote location to pinpoint whether the trouble is in the terminal or its associated data set.

3. The terminal operator will then use the telephone for customer inquiries to the center.

4. The central operator then inputs the inquiry through the console terminal, receives the answerback, and relays the answer by telephone to the remote terminal operator.

B. The central system recognizes that a terminal has failed and looks up a table to find an alternate terminal located at the same remote site or within the same area.

1. The central notifies the alternate terminal that the other terminal has failed.

2. The alternate terminal notifies the failed terminal operator.

3. All messages for the failed terminal are now received by the alternate terminal.

4. All priority messages from the failed terminal are sent from the alternate terminal.

C. If the remote location can tolerate an unavailability of an average of two hours at a time without backup, backup procedures are not necessary.

### 3. 11. 4 RESTART AND RECOVERY PROCEDURES AT THE CENTRAL LOCATION

Restart and recovery procedures depend on checkpoint and restart routines and the methods used for message and file protection. The methods for restart vary with the application and customer requirements, but in general they require the following steps prior to placing the computer center in the online mode after a system malfunction has been corrected.

1. Full system test by online diagnostics.

2. Load the program from disk.

3. Partial display of core storage to ensure that contents of core are correct (make corrections by reloading or keying in from the console).

4. Replace any disk modules that have been altered because of machine malfunction.

5. Initiate the restart program.

6. Check that all disk drives are operating properly.

7. Update tables and areas of core by reading the latest checkpoint information placed on the file.

8. If necessary, notify terminal operators by telephone to switch the terminals back to online mode.

9. Open the lines and test whether each terminal is functioning correctly.

10. Using the information that has been checkpointed, recreate the environment of the

system before it failed.

11. Complete sending all unfinished messages.

12. Update the associated files.

After the line, terminal, or modem has been repaired, restart and recovery procedures must be initiated at the terminal location. A recommended procedure is as follows:

1. Notify the central to place the terminal in test status.

2. At the central, use communication serviceability facilities to receive the test message from the terminal and perform a compare of the test message with its image in core or on file.

3. Send all messages that have been queued at the terminal to the CPU.

4. The terminal sends a message indicating the sequence number of the last message it received.

5. The CPU then sends all messages queued for the terminal indicating to the terminal which messages have been sent to the alternate terminal.

6. Continue normal operation.

If the system is used only for inquiries, the terminal operator inputs his last incomplete inquiry and the CPU would not have any messages queued for the terminal.

### 3. 11. 5 OPERATING MANUAL FOR BACKUP, RESTART AND RECOVERY

For all the procedures that have been described in this subsection, an operating manual is required to enable all the operators to initiate backup, restart and recovery as quickly and as efficiently as possible. During the design phase, the systems engineer or salesman must make the customer aware of this requirement in order that he may include it in his installation plan.

### 3. 11. 6 CONCLUSION

Error control and restart, backup procedures, and restart and recovery procedures are not known to the field in general. For a communications-based system, this subsection, along with the preceding subsection, should enable the user to approach this problem realistically.

## 3. 12 PERFORMANCE ANALYSIS: TIMING AND THROUGHPUT*

A performance analysis phase is required as soon as the configuration selection is completed. This phase helps to analyze how the total system will perform and whether it meets the specified requirements, such as:

- Response time, when given
- Total system throughput
- Availability requirements, if stated
- Systems Assurance criteria for CPU, channel, and units utilization

In the different design phases, the programming systems and hardware configuration have been selected. The system study has helped to establish complete operating procedures and defines the detailed handling of a message under all conditions -- normal and exceptional. Using the programming systems internal logic, a detailed message flow will help:

1. Time the utilization of the different parts, such as:

   - TP program for message control and message processing
   - Access methods
   - Channel utilization
   - DASD utilization

   and assure concurrence, with the objectives of:

   - Total CPU utilization for both batch and T/P
   - Systems Assurance criteria

2. Identify the different queues in the system by queue lengths and sizes, and waiting time.

3. Time a complete message through the whole system to estimate an overall response time (throughput) and also to identify the specific areas that can cause a reduction in throughput.

4. Use the queue sizes to assess worst-case core requirements, either to assure core-size adequacy or provide a method to handle overflow conditions.

Techniques and methods for the performance analysis of the various parts of a communications-based system are covered in the following subsections:

| | |
|---|---|
| Line and Terminal Utilization | Subsections 4. 1 and 5. 2 |
| File Utilization | Subsection 3. 8 |
| Channel Utilization | Subsection 3. 9 |

This subsection of the Handbook deals with total CPU utilization, which includes a channel interference study and processing time analysis.

### 3. 12. 1 CHANNEL INTERFERENCE STUDY

In this study each channel on the CPU is evaluated according to the time it "steals" from the processing time of the CPU to perform data, sense, command, status, and address byte transfers, along with command chaining and interrupts. Channel interference time is added to processing time to determine if the CPU is capable of handling the required processing.

Subsection 3. 9 has shown that the interference on the selector and multiplexor channels due to the addition of a small T/P function is negligible. It is less than 1% on the MPX channel of a Model 30 servicing six 1050 lines loaded 100%, and it is much less than 1% on the selector channel for accessing the files for an inquiry application.

The fact that the T/P function does not add significantly to the interference to the CPU because of the channels does not mean that the application programs (both T/P and background batch) also contribute slightly to the interference. The system designer must calculate the total interference due to the system's I/O operations. The following steps are recommended:

1. Based on the program design, the number of accesses to disk or other I/O operations per transaction type can be estimated.

2. Using the semi-detailed flowcharts developed in the program design, as described in

*For further information, refer to IBM World Trade Systems Bulletin No. 11, OS/PCP Performance and OS/MFT Performance.

subsection 3. 7. 4, indicate the devices that are affected by the various transactions in the flowcharts by associating them with the logical files.

3. Based on message formats and file record formats, estimate the total number of characters transferred to and from the CPU with each I/O operation.

4. Using the Sales and Systems Guide publication, Average Reduction in Processing Capability Caused by Channel Interference in System/360 Models 30, 40 and 50, Form Z20-1180, develop the interference caused by each device for each transaction type.

5. Multiply the interference caused by each device for each transaction type by the number of transactions of this type occurring during a peak period.

   (interference due to device) x (no. of transactions)
   per transaction type

6. Add all the interferences caused by the various transaction types to arrive at the total interference due to a particular device.

Total device interference = (transaction type I interference) + (transaction type II interference) + . . .

7. Separate the various I/O devices by channel, such as MPX, selector channel 1, selector channel 2, and selector channel 3 and obtain the total interference caused by each channel.

8. Add the interference caused by each channel to obtain the total interference caused by the channels to the CPU.

9. Add a 20% contingency to the total interference. If the total interference including the 20% contingency exceeds the recommended 60%, a faster and more powerful CPU or a review of the system design is required.

### 3. 12. 2 PROCESSING TIME ANALYSIS

This analysis results in the total CPU time required for executing instructions during the peak period. The analysis may be considered in two phases:

1. Estimate the number of instructions required to process each message type;

2. Estimate the instruction time required to perform the line control functions not accounted for, on a per message basis, such as processing polling interrupts, etc. If applicable, hardware interlock time should be estimated and included.

The message mix during the peak time period is then used to determine the total processing time.

Total processing time includes the time to execute the macro instructions for BDAM, BTAM, and other O/S modules and the instruction time used by the application program to process the required transactions during the peak time period.

The major steps required to perform a processing time analysis are:

1. Using the semi-detailed flowcharts developed in the program design, as described in subsection 3. 7. 4, estimate the number of instruction cycles each function block in the flowchart would take per transaction. Indicate the number of cycles on the chart.

2. From IBM Operating System/360 Performance Estimates, Form Z28-6552, obtain the time required for each macro operation for initiating a read or write operation to an I/O device.

3. Indicate on the flowcharts the macros required for the I/O operations.

4. Estimate the number of times each of the macros must be executed per transaction.

5. Obtain the total macro time spent per transaction.

6. Obtain the total number of application instruction cycles per transaction.

7. Multiply the instruction cycles by the average execution time per instruction. Conservative figures for this operation are 35 μs for the Model 30; 20 μs for the Model 40; and 12 μs for the Model 50. This will give the total time spent by the application on the processing of a transaction.

8. Add the total processing time per transaction to the total macro time for the same transaction to obtain the total processing time spent for the particular transaction. Apply this same operation to all transaction types.

9. Multiply the total processing time spent for a transaction by the number of these transactions occurring during the peak time period. Apply this method to all transaction types. This will give the total processing time spent during a peak time period for each transaction type.

10. Add all the processing times per transaction type during the peak time period to obtain the total processing time spent by the CPU.

11. Add a 20% contingency to the total processing time. If the total processing time, including the 20% contingency, exceeds the maximum permissible load (60% recommended) on the CPU, change of the CPU type or review of the system design is required.

### 3. 12. 3 TOTAL CPU UTILIZATION

Once the T/P part is judged satisfactory, total CPU utilization (including batch) must be estimated to see if it fits the capability of the CPU selected.

Add the total interference time to the total processing time to obtain total CPU utilization. Calculate what percent of the CPU is utilized during the peak period. If the CPU is utilized over the recommended 60% during this time, a faster and more powerful CPU is required and/or a review of the system design is necessary in order to determine where changes can be made in programming and I/O units so as to decrease the utilization. If the user wishes to utilize the CPU beyond the recommended 60%, he is advised to make sure that all areas in his design have been considered thoroughly.

### 3. 12. 4 TOOLS FOR PERFORMANCE ANALYSIS OF A SMALL CBS

Several tools for analyzing the actual limitations which will help define a small CBS and separate the actual CPU utilization for the foreground (communications application program) and the background programs, may be used to derive the throughput capability of the communications foreground programs under the DOS or MFT programming systems environment.

These formulas can be used by simply inserting the actual values for the user's system. The set of formulas given will help to assess quickly the potential limitations of the user's system.

A. Definitions of Parameters Used in the Formulas and Curves.

1. A transaction is defined here as the input message and its answer from the CPU. The parameters defining the system are:

   a. Transaction (in + out) length in total number of characters = a.
   b. Number of communication lines attached to the system = L.
   c. Effective speed of transmission on the line in characters per second = S.
   d. Maximum line loading = Y.

2. The parameters defining the processing are:

   a. Application processing time per transaction = P msec.
   b. Number of accesses to file per transaction = n read and/or write commands.
   c. Execution time per read/write macro to file = f msec.
   d. Execution time per BTAM read/write to line = t msec.
   e. Access time to the record or file response time (as defined in File Organization, subsection 3. 8. 4) = w msec.

B. Assumptions

The assumptions made on the actual CBS environment are those for a small CBS.

1. The communications program can handle only one transaction at a time, while others can be assembled for further handling (sequential processing).

2. During a WAIT macro issued after a read or write file, only a program with lower priority can use the CPU, except for a reinitiation of polling by the supervisor after a negative answer to polling (refer to Programming, subsections 2. 6 and 3. 7).

3. There is only one BTAM read and one BTAM write per transaction.

4. The queue time for the transaction as it waits in the central system to be processed is negligible. (Refer to Queuing, subsection 3. 4. )

5. The channel interference time caused by the input and output messages from

and to the 2701 and 2702 on the MPX channel is negligible. (Refer to subsection 3. 9, Channel Interference. )

6. The communications application program cannot be busy more than 60%.

7. The communications lines cannot be loaded more than 60%.

## C. Constants

Constants that can be used for processing parameters are shown in Figure 58.

The two major variables are P (application processing time) and n (the number of file accesses). The number of file accesses has already been defined for an inquiry and updating operation. The user must decide if this is a reasonable number for his particular application; and, if not, increase or reduce it accordingly.

The communications application program will define the application processing time (P). The program can be divided into two major sections, message and line control, and message processing.

Message and line control is assumed to include the following:

1. Header analysis - input
   - a. Verifying the message begins with a proper character sequence
   - b. Verifying the header portion of the message ends with a proper character sequence
   - c. Recognize the message type
   - d. Verify origination of terminal code
   - e. Check the input message sequence number
2. Setting up proper information for a read or write to a file
3. Code translation for input and output message
4. Buffer manipulation for input and output

| PARAMETER | 30 | 40 | 50 | REMARKS |
|---|---|---|---|---|
| P | 35 μS | 20 μS | 12 μS | THESE ARE AVERAGE INSTRUCTION TIMES FOR EACH INSTRUCTION CYCLE EXECUTED IN THE APPLICATION PROGRAM . |
| N | 6 | 6 | 6 | THIS IS FOR AN INQUIRY AND UPDATE MESSAGE WITH THE FOLLOWING OPERATION:<br>1 . WRITE INPUT INQUIRY ON PRÒCESSING QUEUE<br>2 . READ INPUT INQUIRY FROM PROCESSING QUEUE<br>3 . READ RECORD FROM INFORMATION FILE<br>4 . WRITE UPDATE RECORD TO INFORMATION FILE<br>5 . WRITE ANSWERBACK TO OUTPUT QUEUE<br>6 . READ ANSWERBACK FROM OUTPUT QUEUE |
| F | 20 MS | 11 MS | 5 MS | THESE ARE AVERAGE TIMINGS FOR READ/ WRITE MACROS USING SPS (REFERENCE SUBSECTION 3. 8. 4). IT IS ASSUMED HERE THAT THE DOS MACROS HAVE THE SAME TIMINGS . |
| T | 7 MS | 4 MS | 2 MS | THIS IS AN AVERAGE FOR A BTAM READ/ WRITE MACRO USING DOS WITH A MAXIMUM OF 10 MS AND A MINIMUM OF 3.4 MS FOR A MODEL 30. IT IS ASSUMED HERE THAT THE BTAM/ SPS MACROS HAVE THE SAME TIMINGS . |
| W | 150 MS | 150 MS | 150 MS | THIS IS THE FILE RESPONSE TIME FOR THE 2311–2314 (REFER TO SUBSECTION 3. 8. 4). |

Figure 58. Constants Used for Processing Parameters

5. Prepare-to-send message - output

Message processing consists of a program that uses the text of the input message to perform such operations as preparing answerbacks to inquiries, updating files, and other typical data processing functions. The message and line control program's execution can be reduced to the number of instruction cycles per character, which is based on a constant for a message of any length plus a variable number of instructions based on the message length due to such routines as code translation and buffer manipulation. This can be expressed in the following formula:

(1) Total instruction cycles = constant per message + number of characters x (instructions/character)

(2) Number of instruction cycles/character = $\frac{\text{total instruction cycles}}{\text{message length}}$

For a small CBS, a conservative rule of thumb for instruction cycles per character for message and line control can be based on 7740 experience. To perform the message and line control function within the 7740, including setting up for reads or writes to a file, took a range of 15 to 25 instruction cycles per character. Since the constant number of instructions per message is significant only for short messages, the following rules apply:

| Message Length | Instruction Cycle/Character |
|---|---|
| Less than 50 characters | 30 |
| Less than 100 characters | 25 |
| Equal to or greater than 100 | 20 |

$$\text{Message length} = \frac{\text{input message length} + \text{created output message length}}{2}$$

Refer to Figure 59 for an example of processing by message length for message and line control for the Models 30, 40, and 50.

Therefore, the total communications application processing is as follows (Figure 59):

| MESSAGE LENGTH IN CHARACTERS = $\frac{\text{IN + OUT}}{2}$ | TOTAL TIME IN MS | | |
|---|---|---|---|
| | MODEL 30 | MODEL 40 | MODEL 50 |
| 10 | 11 | 6 | 4 |
| 50 | 44 | 25 | 15 |
| 100 | 70 | 40 | 24 |
| 300 | 210 | 120 | 72 |

Figure 59. Timings for Message and Line Control

D. Calculations for Obtaining Throughput

1. Total transaction time:
For each transaction handled, the communications program is busy (either processing or in a wait status) for a time (T) equal to the result of formula (3) in microseconds.

T = (processing time) + (number of accesses) x (macro time) + number of accesses) x (file response time) + 2 x (BTAM macro time)

(3) $\boxed{T = P + n \times f + n \times w + 2t = P + n\,(f + w) + 2t, \text{ microseoonds}}$

The total time used by one transaction within the CPU is:

Total transaction time = (communication program busy time) + (queue time) + (a portion of channel interference)

As was mentioned previously, queue time and channel interference time can be disregarded for a small CBS. Therefore:

Total transaction time = T.

Total transaction time is also known as the internal response time, which is defined for a small CBS as being equal to or greater than 2 seconds.

2. Maximum number of transactions handled by the network:
As for the maximum number of transactions of the same type per busy hour being handled by the network (which we will call $m_1$), they are given by:

$m_1$ = max. number of transactions

$$= \frac{(\text{no. of lines}) \times (\text{effective speed}) \times (3600 \text{ sec./hr.}) \times (\text{line load})}{\text{transaction length (In + Out)}}$$

(4) $\boxed{m_1 = \frac{L \times S \times 3600 \times Y}{a}}$

where Y is the permissible line load as defined in Traffic Analysis and Network Design, subsection 3.5.

3. Maximum number of transactions allowed by the communications program

Based on these assumptions, we know that the communications program cannot handle more than one transaction at a time. Therefore, in one hour, the communications program, if busy 60%, will be able to handle only the maximum number of transactions $m_2$, equal to:

$$m_2 = \frac{(3600 \text{ sec./hr.}) \times (1000 \text{ ms}) \times (60\% \text{ busy})}{\text{Total transaction time}}$$

$$(5) \quad \boxed{m_2 = \frac{3600 \times 1000 \text{ msec} \times .60}{P + 2t + n\,(f + w) \text{ msec}}}$$

Based on the 2-second internal response time:

$$m_2 = \frac{3600 \times 1000 \times .60}{2 \times 1000} = 1{,}080 \text{ transactions/hour}$$

4. Maximum number of lines

Given a transaction length "A" and an effective speed of transmission "S", the maximum number of lines loaded at Y which can be handled by the communications program is such that $m_1 = m_2$, or:

$$\frac{3600 \times S \times L \times Y}{a} = \frac{3600 \times 1000}{P + 2t + n\,(f + w)}$$

$$(6) \text{ or } \quad \boxed{L = \frac{1000 \times (a) \times .60}{[P + 2t + n\,(f + w) \quad \times S \times Y\,]}}$$

which defines the maximum number of lines which can be handled under these conditions.

The limitations for examples of systems given in the Handbook, using an average transaction length of 300 characters, are shown in Figure 60.

As transaction length decreases, the maximum number of lines decreases.

As effective speed increases, the maximum number of lines decreases.

As total transaction time increases, the maximum number of lines decreases.

E. Estimated CPU Utilization for the Communications Application Activity

Other aspects which might introduce systems limitations are:

1. Channel loading added by the communications application file accesses to the same channel and MPX channel interference.

2. Load of the CPU due to the communications program.

In computing this load, we find that for each transaction the CPU is used for P + 2t + nf + transfer time (channel utilization) or total transaction time, plus channel utilization, less the file response time.

Thus, the total load on CPU for "m" transactions per hour is:

(total transactions) x (total transaction time - total file response time), which equals m (P + 2t + nf) msec/hour. If we omit transfer time (channel utilization), the total load on the CPU (C) in percent would be:

$$(7) \quad \boxed{C = \frac{m\,(P + 2t + nf)}{3{,}600{,}000} \times 100}$$

For example, using a Model 30 with the following parameters:

n = 3 accesses, P = 30 msec (processing time)<br>
f = 20 msec (file macro time)<br>
2t = 14 msec (2 x BTAM macro time)

The number of transactions to be handled is: m = 10,000 per busy hour; the total CPU utilization time for the communications part is:

10,000 (30 + 14 + 60) = 1,040,000 msec.

| SYSTEM | ACTUAL LINE SPEED | EFFECTIVE SPEED | MAXIMUM NUMBER OF LINES |
|---|---|---|---|
| 1050-2740 | 14.8 CPS | 10 CPS | 15 |
| 2260 | 120 CPS | 80 CPS | 2 |
| 2260 | 240 CPS | 180 CPS | 1 |

Figure 60. Limitations on Line Speeds

or the load on the CPU

$$\text{in \%} = \frac{\text{utilization of CPU by communications} \times 100}{\text{total time CPU available for use}}$$

$$C = \frac{1,040,000 \times 100}{3,600,000} = 29\%$$

For detailed examples of program timing and rule of thumb, refer to Section 4 on the 1050, and the examples in Sections 4, 5, and 6.

F. Example Showing the Use of All the Preceding Tools

The system to be analyzed is based on a Model 30 and has the following parameters:

- Input + output transaction length = a = 100 characters
- There are 500 transactions during the peak hour
- Message processing time based on 1,000 instruction cycles
- Effective speed for the 1050: S = 10 cps
- Average number of accesses: n = 6
- Customer requires more than 2 seconds internal response time

1. The first step is to find out what the processing time (P) is for the transaction.

a. P = (message and line control) + (message processing)

b. From Figure 59, the time for message and line control on the Model 30 for a 100-character transaction is 70 ms.

c. From Figure 58, the average instruction time for a Model 30 is 35 µs.

d. Total (message processing) = instruc. cycles x 35 µs = 35 ms

e. P = 70 + 35 + 105 ms

2. The next step is to find out the total transaction time (T).

a. Using formula (3) and Figure 58, we find for a Model 30:

f = 20 ms (file macro time)

t = 7 ms (BTAM macro time)

w = 150 ms (file response time)

b. From formula (3):

$$T = P + n\ (f + w) = 2t$$

$$= 105 + 6\ (170) + 2\ (7)$$

$$= 1.139 \text{ sec.} = \text{internal response time*}$$

3. The third step is to estimate the maximum number of lines for the system.

Since the internal response time is below 2 seconds, the simplified formula (8) can be used:

$$(8)\ \boxed{L = \frac{a}{25}} = \frac{100 \text{ char.}}{2 \times 10} = 5 \text{ lines}$$

4. The next step is to find out the maximum number of transactions that can be handled by the network.

a. Using formula (4):

$$m_1 = \frac{5 \text{ lines} \times 10 \text{ cps} \times 3600 \times .60}{100}$$

$$= 1,080 \text{ transactions/hour}$$

b. This is within the customer requirements of 500 transactions/hour.

5. Finally, the user must make sure that the CPU can process 500 transactions/hour. Using formula (5), it has been shown that with a 2-second internal response time, 1,080 transactions can be processed.

## 3. 12. 5 CONCLUSION

This subsection has offered to the user various methods and techniques for evaluating his system. It has also further developed the limitations (definition) of a small CBS.

*This is well within the customer requirements of over 2 seconds and within the requirements of a small CBS.

## 3. 13 WTC SYSTEMS ASSURANCE

### 1. What Systems Assurance Is

Systems Assurance is a program intended to provide, prior to proposals, an independent review of new and complex system configurations. It was initiated mainly to assure both the customer and IBM that a specific system configuration can satisfy the customer's requirements.

As a secondary objective, the program determines whether the performance characteristics of the system are presented to the customer in a form consistent with IBM practices.

### 2. What Systems Require Systems Assurance

The specific system configurations requiring Systems Assurance review and approval prior to the proposal are indicated in the Designated List of the WTC DP Sales Manual, General Information, Marketing Policies, pages 1 and 2. The Designated List, as it stands currently, includes the following system configurations:

- Online Tele-processing systems
- Data acquisition and control systems
- Interconnected systems
- Special configurations (time-sharing, 2260, etc.)

As new products are released, announcement letters will indicate whether they are subject to S/A review and approval.

In addition, S/A approval is required for all orders of components that convert an installed or on-order system into one of the designated configurations; or for subsequent changes in configurations which were previously approved under this program.

### 3. How S/A Activity Is Performed

As soon as a study is undertaken which is expected to result in a configuration subject to Systems Assurance, the field authority responsible must register the system with the local Systems Assurance function (this can be, depending on local arrangements, a formal Systems Assurance department, a Systems Assurance coordinator, or the country SE manager).

Following this registration, Systems Assurance requests all necessary system documentation in order to indicate proper review activity.

Depending upon the system's complexity, one or more pre-review meetings may be held, to make sure that the design work is heading in the proper direction. During these meetings, S/A will also provide technical support, if necessary, by consulting with other specialized functions of IBM.

As soon as the final system documentation is available, a formal review meeting is held. Field representatives are normally present, as their presence should facilitate the review procedure and provide them with an opportunity to expand their understanding of the type of system.

After the review meeting, an S/A review report is issued by the S/A function, with the following outline: system description, design assumptions, review comments, conclusions (system approved, not approved, or conditionally approved).

### 4. What System Documentation Must Be Submitted

The responsible sales personnel must submit the following documentation prior to the final review meeting:

(1) Copy of the customer's request for proposal or written speficiations, if they exist;

(2) Copy of the IBM proposal and other written documents that will be presented to the customer;

(3) Copy of approved RSDP's, if any;

(4) Copy of Branch CE Manager's Letter on the maintenance of the system (see Review Guide);

(5) Detailed answers to the Review Guide.

### 5. Review Guide

The Review Guide is the most important source of review information. It is intended as a check list of those aspects of the system on which the S/A review will be focused, and should be answered in detail as part of the required review documentation. Particular attention should be paid to the amount, level, and appropriateness of the required IBM support.

The documents submitted to the S/A may sometimes include part of the information requested in the Review Guide; in such cases it suffices merely to reference separate documents from the appropriate sections of the Review Guide.

### 6. S/A Follow-Up

An S/A follow-up is necessary for all systems given conditional approval to make sure that the required action is actually taken.

In addition, any modification to a configuration or an application subsequent to S/A review and approval requires a supplement to the S/A review. The branch office is responsible for advising S/A of such changes before any commitments are made to the customer.

Similarly, if any variation occurs concerning performance or availability of both hardware and programming related to the system, the local SE management is responsible for initiating proper corrective action requesting S/A concurrence, if necessary.

### 7. S/A Review as a Technical Support Function

While making a review of the system, S/A also provides valuable technical support to sales representatives, by making its own knowledge available and by consulting with other technical functions of IBM. It is therefore to the interest of the field representatives to give timely and cooperative attention to the S/A program, both to improve the quality of the system and of the proposal, and to obtain S/A approval without unnecessary delay.

## SECTION 4.

## 1050 SYSTEM DESIGN

The 1050 system consists of a control unit which performs the control function for different types of input and output units, such as paper tape and card reader, paper tape and card punch, and typewriter-keyboard (see Figure 61). Only a typewriter-keyboard can be connected to the control unit of the 2740, which operates with the same techniques as the 1050. The 1050, as well as the 2740, has a transmission speed of 14. 8 characters/second.

The section on Terminal and Line Control describes the use of the line facility and how communication between a computer and terminals takes place.

The section on Terminal and Line Loading explains the reasons why the actual throughput of the system is less than the transmission speed of 14. 8 characters/second. Examples are given to derive rules of thumb.

Besides the Sales Manual and configurators for the selection of the front-end hardware, the section on Terminal and Multiplexor Configuration will be helpful.

A great many systems consist of up to about six lines with one 1050 or 2740 terminal per line. For this type of configuration, we have provided a special approach in the section on Simplified Procedures for the Design of a CBS with 1050 or 2740 Terminals.

For a better understanding of the whole 1050 System section, in connection with general subjects like file organization, CPU, channel loading and interference, CPU analysis, queuing, etc. , see the 1050 Example.

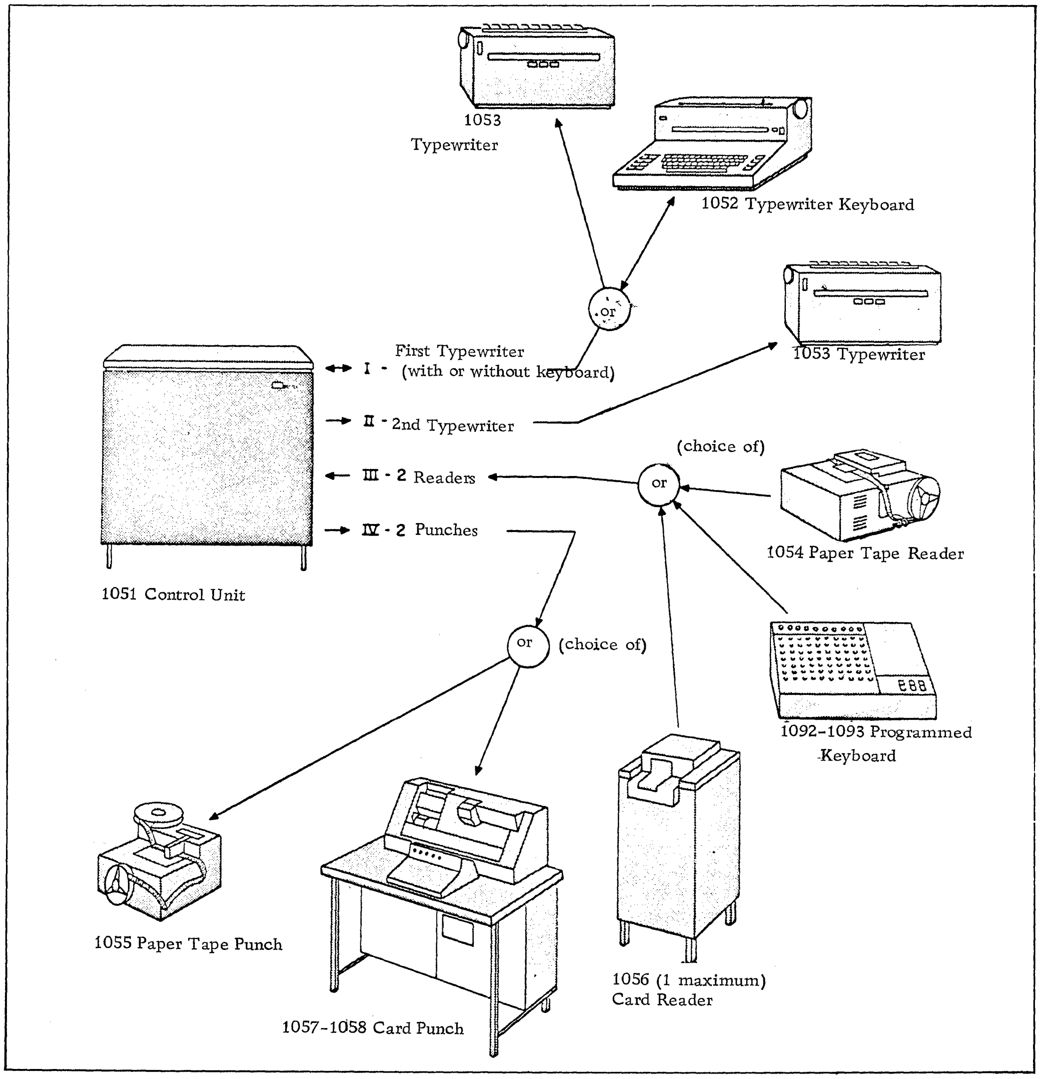


Figure 61. Maximum Configuration of 1050 System

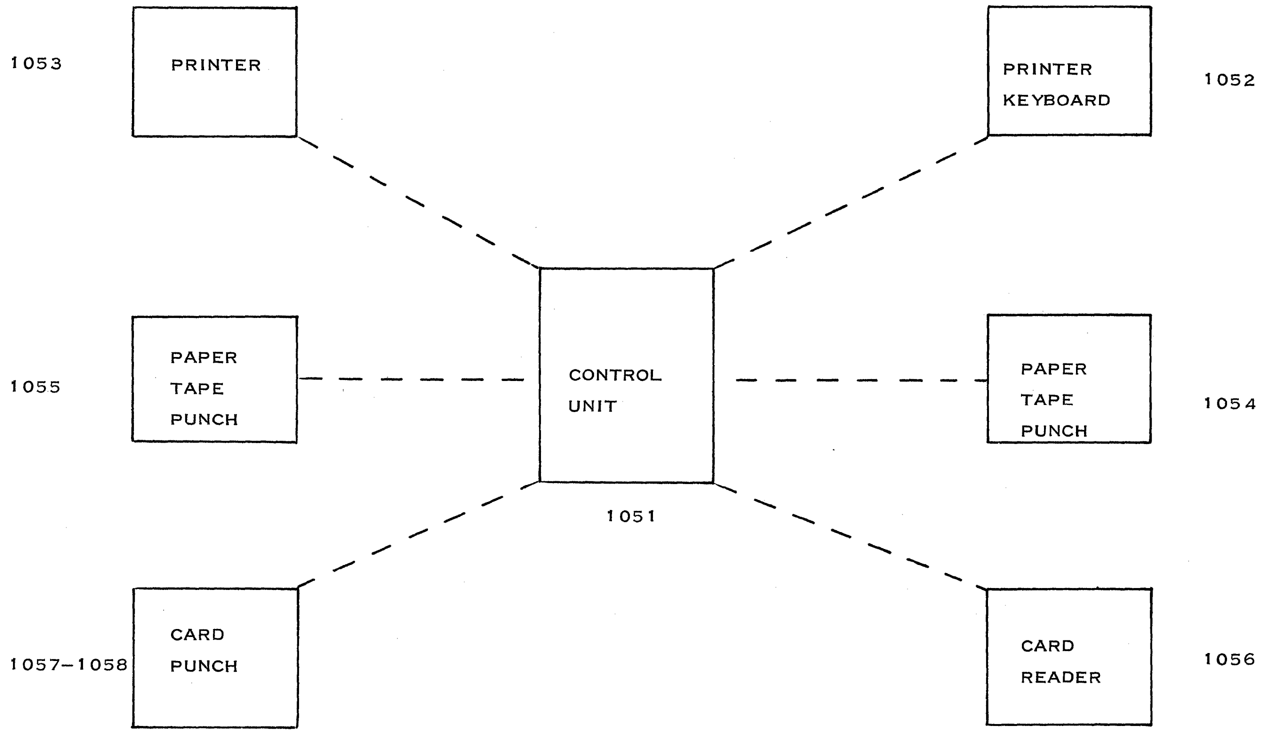
## 4. 1 TERMINAL AND LINE CONTROL

### 4. 1. 1 TERMINAL CONTROL

To identify the individual 1050 systems sharing a line, each station is assigned one of the 26 characters of the alphabet to denote its unique station identification.

A secondary identification character can be assigned to each station (system) in a designated group to permit group addressing (simultaneously addressing all stations). A second numeric character in the address is used to select the desired component (see Figure 62).

If this second character defines an output component, the addressing sequence is part of an addressing operation. If it defines an input component, it is part of a polling operation.



COMPONENT SELECT CODES

| ADDRESSING | | | POLLING | |
|---|---|---|---|---|
| 1 | PRINTER | 1 | 5 | KEYBOARD |
| 2 | PRINTER | 2 | 6 | READER |
| 3 | PUNCH | 1 | 7 | READER |
| 4 | PUNCH | 2 | 0 | ANY ONE INPUT COMPONENT |
| 9 | ANY OR ALL OUTPUT COMPONENTS | | | |

Figure 62. Addressing: Identification Characters

### 4. 1. 2 ERROR CONTROL

Data transmission must be error-controlled. The current way of doing this is to use a cross-check, known as VRC (Vertical Redundancy Check), plus LRC (Longitudinal Redundancy Check). Each character receives an additional "check" bit whereby the total number of 1's is even or odd, depending upon an agreed rule. Further, the same is done for all bits of the message located at the same rank which generates an additional LRC character. This kind of control is very powerful: whenever there is a disagreement between the sending and the receiving terminals on VRC and/or LRC, an automatic retransmission can be done, if the line connection feature is installed. After the third unsuccessful transmission, an alarm can be given to the operators for further investigation of the data terminal, modem or line.

Line Control Signals

| | | |
|---|---|---|
| Ⓒ | End of Transaction (EOT) | 1051 control mode |
| Ⓓ | End of Address (EOA) | 1051 text mode<br>positive response to polling |
| Ⓑ | End of Block (EOB) | LRC character follows |
| Ⓨ | Positive Response | to addressing |
| Ⓝ | Negative Response | to addressing (not ready)<br>to polling (no data) |

### 4. 1. 3 ERROR CONTROL SIGNALS

| | | |
|---|---|---|
| LRC ANSWER | Ⓨ | positive |
| | Ⓝ | negative (error) |
| | Ⓓ | positive + answer to inquiry |

The line control signals monitor the flow of data over the line to avoid line contention. The coded character Ⓒ , recognized as EOT, is used:

- by the computer to place all terminals in the receive mode;
- by the computer and the terminals to indicate that there is no more data to transmit.

The coded character Ⓓ , recognized as EOA, indicates that the following character will be the first of the block and starts the LRC.

The coded character Ⓑ , recognized as EOB, indicates that the block is terminated. EOB is immediately followed by the LRC character generated at the transmitting point, for purposes of comparison with the LRC character obtained at the receiving point.

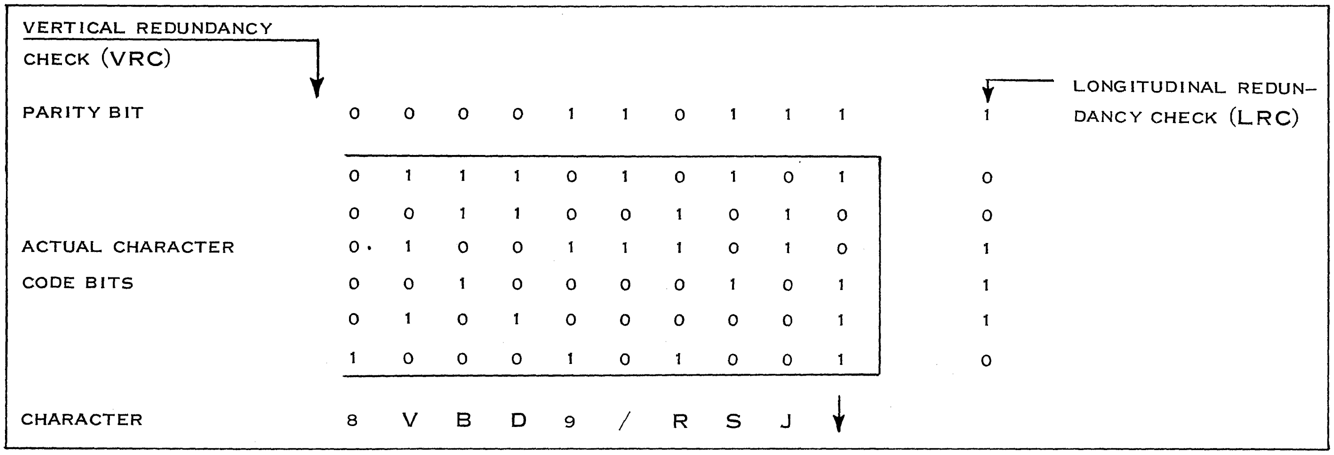
The coded character Ⓨ , is a positive answer to:

- polling or addressing;
- LRC.

Note: If any other code combination is received due to transmission error, it will be considered as a negative answer. The coded character Ⓝ is a negative answer to:

- polling or addressing;
- LRC.

The coded character Ⓓ is used in the case of a terminal sending an inquiry to a computer. The Ⓓ is sent from the computer to the terminal, indicating that the inquiry was correctly received.


VERTICAL REDUNDANCY CHECK (VRC)
LONGITUDINAL REDUNDANCY CHECK (LRC)

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| PARITY BIT | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| ACTUAL CHARACTER | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| CODE BITS | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| CHARACTER | 8 | V | B | D | 9 | / | R | S | J | ↓ | |

Figure 63. Vertical Redundancy Check and Longitudinal Redundancy Check

### 4. 1. 4 ADDRESSING

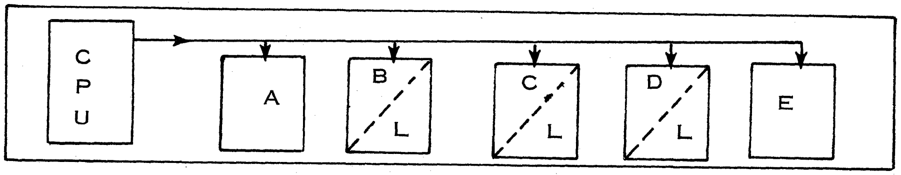


Figure 64. Multi-drop Configuration: Addressing

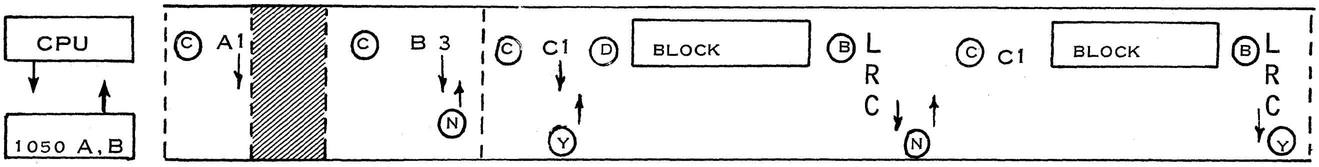


Figure 65. Addressing: Systems A, B, and C

Assume that the processor (shown in Figure 64) has messages for terminals A, B, C, and D. The CPU starts the transmission sequence by sending Ⓒ (to clear the line and causes all ready terminals to monitor the succeeding characters), and then A 1 (to address printer 1 of the 1050 A). If the terminal does not answer during a 3-second time-out, it is considered inoperative (i. e. power off, drop-line dead, etc. ). In some cases, the terminal is addressed a specific number of times before being considered inoperative.

After the time-out, the CPU resumes transmission by sending Ⓒ , B 3, which addresses the paper tape punch of the 1050 B to receive a message. If the paper tape punch is inoperative (for example, out of tape), the 1050 B replies with Ⓝ , the negative response.

The CPU next addresses 1050 C by Ⓒ, C1. Let us assume that the 1050 C replies with a Ⓨ (positive response, ready to receive). The CPU resumes with Ⓓ (to start the text LRC count), followed by the message text ("block" in Figure 65), followed by Ⓑ (EOB, End-of-Block, for ending the LRC count and indicating that the succeeding character is the LRC), and then the LRC itself. If the 1050 C replies with Ⓝ to signify that it detected an LRC error in transmission, the CPU retransmits the Ⓓ message, Ⓒ , LRC sequence. The 1050 C replies with a Ⓨ indicating successful transmission.

For terminal D, it is intended that the message be printed and punched simultaneously. Each component (D 1 and D 3) is addressed and gives a positive answer, as shown in Figure 66. Then transmission takes place as previously indicated.

Then follows a message of interest to terminals B, C, and D (for example, each of them is used for similar activities: plants, branch offices, laboratories). The CPU uses the group addressing character L and sends Ⓒ L 9, which means any or all output components of terminals B, C, and D. A positive answer is given by one terminal. In the example shown in Figure 66, D is chosen as it is the most remote system on the line. Then transmission takes place, and the positive answer to LRC is also given by terminal D.

The CPU tries again to reach terminal A, which is now ready and able to receive its message.

The CPU requests again the paper tape punch of terminal B to accept a message. The punch is ready to receive and the CPU sends the message, which is punched at terminal B location.

Whenever a message must be received by all the terminals, a special "broadcast" character, /, is used. Ⓒ /9 means that all systems must receive

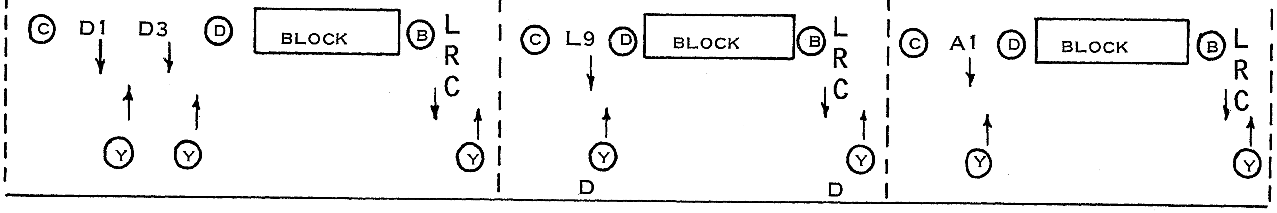


Figure 66. Addressing: System D

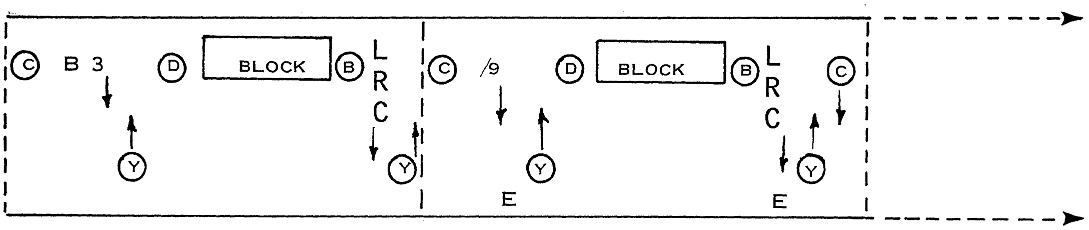


Figure 67. Addressing: Positive Answer for Broadcast

the same message. The positive answer to broadcast addressing (as well as the positive answer to LRC) is given by one selected terminal, which is usually the most remote system at the end of the line. An example of broadcast message control is shown in Figure 67.

### 4. 1. 5 POLLING

To send messages from any terminal to the CPU, as shown in Figure 68, line contention must be avoided. Line contention would appear if every terminal were allowed to transmit at any time, without central supervision. The polling procedure is based on two criteria: line access must be given to every system as often as possible; average traffic of every system is taken into consideration. For instance, if the average traffic of every terminal is approximately the same, the polling sequence would be:

A B C D E A B C . . .

If terminal C has twice as much traffic as any other, the polling sequence can be adjusted to:

A B $\underline{C}$ D E $\underline{C}$ A B $\underline{C}$ D E $\underline{C}$ A . . .

Starting with Ⓒ for the same purposes as previously indicated in addressing, the CPU polls terminal A. If after three seconds the CPU does not receive an answer from terminal A, it polls B, as shown in Figure 69. The negative answer Ⓝ indicates that B has no traffic for the CPU. Now polling terminal C, where there is waiting traffic, there is no need to send a positive answer since, at that point of time terminal C has permission to communicate with the CPU. Consequently, terminal C immediately transmits Ⓓ (End of Address), followed by the first block of its message, End of Block Ⓑ , and the LRC character. The first block is correctly received by the CPU. Then block 2 is sent and repeated. A positive answer being received after the second transmission, terminal C indicates that it has no more traffic by sending the End of Transmission character Ⓒ , which allows the CPU to poll other systems in the polling list.

When polling terminal D, as indicated in Figure 70, it appears that D wants to send a message to E on the same line (L 1). L 1 indicates the same line. (It should be noted that D could very well reach any other terminal connected to another multi-drop line, L2, L3, L4, L5, etc.) The preamble Ⓓ L 1 E 1 will be recognized by the CPU as a retransmission request of the message which follows. Having received the Ⓒ (End of Transaction) from terminal D, which has no more traffic, the CPU immediately addresses the message to E 1 (printer of terminal E)

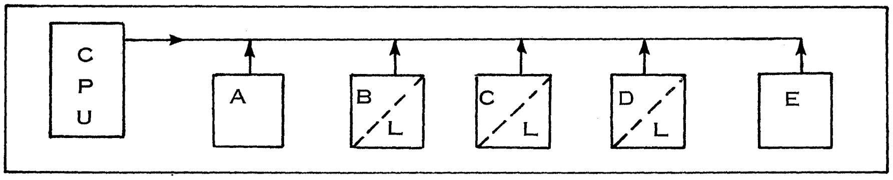


Figure 68. 1050 Multi-drop Configuration: Polling

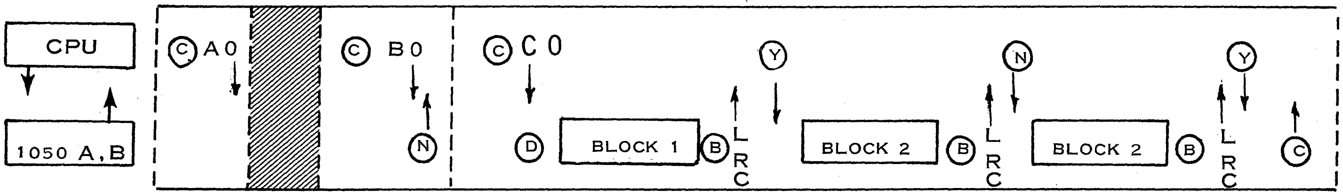


Figure 69. Polling: Systems A, B, and C

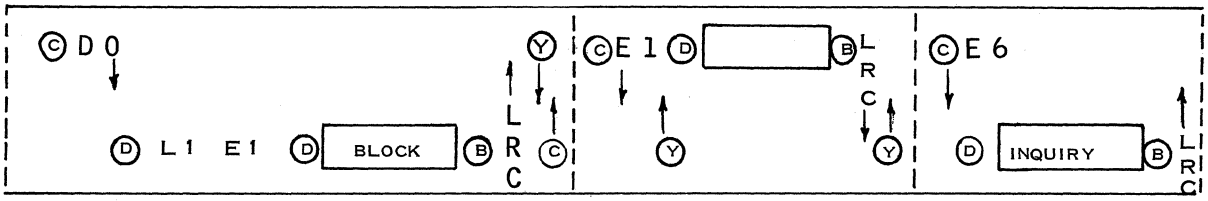


Figure 70. Polling: System D

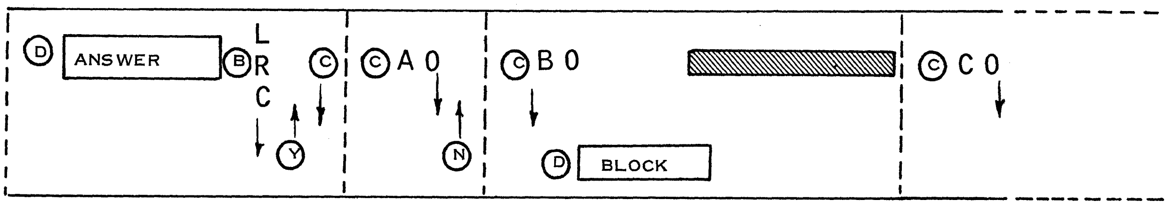


Figure 71. Polling: Inquiry

and then polls E.

System E has an inquiry for the CPU. The inquiry message is sent like other messages, but in the message the CPU will detect the inquiry. In this case, the positive answer to the LRC control is ⓓ, as shown in Figure 71, immediately followed by the answer to the inquiry. After positive acknowledgement of correct reception of the answer by System E, the CPU starts a new polling sequence.

System A, now ready, has no traffic. System B has traffic and starts to transmit. During the transmission there is a malfunction (equipment, modem, or line failure) and the CPU does not receive the End of Block. After a time-out of 20 seconds, the CPU resumes the polling sequence with System C, D, and so on.

## 4. 2 TERMINAL AND LINE LOADING

### 4. 2. 1 GENERAL

This chapter examines the reasons why the actual throughput of the 1050 system is less than the transmission speed of 14. 8 characters per second. The actual throughput is called the "effective speed".

As the effective speed is very important for calculating the terminal load, we shall describe the influence of this on each of the input/output components. However, we know that in the design phase it often happens that the customer cannot supply enough information to make an exact calculation. In such cases, we have to use rules of thumb and instructions for applying them effectively.

To calculate the terminal load, the number of data characters travelling along the line must be the starting point. We have to take into account the following additional characters:

- System control characters
- Device control characters, including idle characters.

Moreover, we have to allow for the time during which no characters are travelling along the line. During this time, neither the line nor the terminal is available for other use. Examples are:

1. Necessary time (turnaround time) to switch a unit from send to receive status, or vice versa.
2. Delay time to execute the different functions within the terminal.
3. Operation time necessary to insert forms feeding and ejecting cards, skipping, releasing, etc.

Furthermore, there is the time necessary to execute retransmissions. If there is a transmission error and the automatic retransmission feature has been installed, the last block of information is repeated automatically. This is, of course, not valid for any keyboard transmission other than the 1092 and 1093, as after an error indication on a keyboard, the data block has to be manually rekeyed. To calculate the time for automatic retransmission, we have to know the block length and the error rate of the line. Let us suppose we have an overall block length of n characters and the line quality is one error per $10^{X}$ characters. We have to retransmit approximately n characters every $10^{X}$ characters.

In practice, it can be estimated that a good-quality leased line has an error rate of one error per 100, 000 characters transmitted. Thus, if we assume data blocks of 50 characters, only approximately one block in every 2, 000 requires retransmission. Obviously, this is . 05 percent and quite negligible.

Even if we assume much worse conditions (which would not normally prevail on a line) such as one error in 10, 000 characters with 100 characters per block, we would still have only about one retransmission for every 100 data blocks. Then our retransmission rate is still only 1%, which, in view of the usual lack of precise volume estimates, is quite insignificant and can be ignored.

### A. System Control Characters

These characters, which are used to control the traffic on the lines, have already been described in detail in subsection 4. 1 on Terminal and Line Control. We now want to consider their influence on timing.

First, however, we have to define what we consider the time necessary to address or poll a terminal and what time we need for the error-checking procedure called "end of block time".

The definitions indicated in Figure 72 have been chosen so that all line control characters are included. This simplifies the throughput calculations

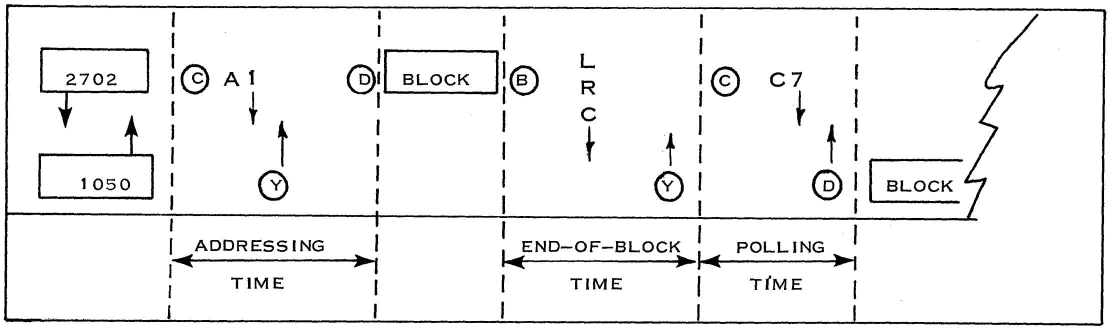


Figure 72. Addressing, End of Block, and Polling Times

of the system. Now we need only add the time for the transmission of the message itself.

The effect of negative responses to addressing, polling or end of block sequence will be ignored.

To produce timings, we have considered a 1050 system connected to a 2702 transmission control unit.

The general procedure is as follows:

The first WRITE command will cause a one-character delay with all mark bits, including the start bit (this in effect is a delay of one-character time, or 67.5 ms). This is the turnaround time, which gives the line time to stabilize. At the end of this character, the clear-to-send condition is tested. If it is not on, another such padding character is sent, at the end of which the clear-to-send is again tested. This process is repeated until the clear-to-send is on.

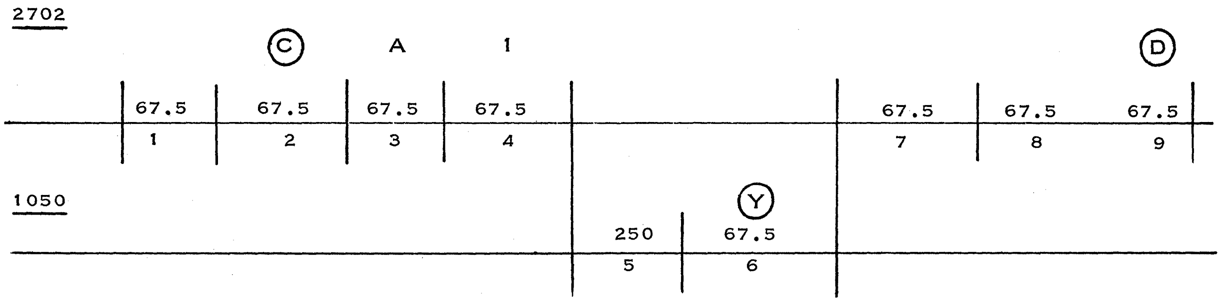
The clear-to-send delay of the IBM line adapter or the IBM 3976 Model 1, in full- duplex operation, is 15-25 ms, and hence only one padding character will be sent.

When the clear-to-send is on, the polling or addressing sequence is sent. At the end of this sequence, the WRITE command ends and a READ is issued. The 2702 then waits 3.0 seconds for an answer (the 1050, when.ready, sends its answer after 250 ± 50 msec). If no answer is received within this time, the program will decide whether to resend the polling/addressing sequence or abandon this terminal and poll/address a second terminal. Whatever the case may be, the 2702 will wait until the line is quiet for one whole character-time before ending the READ command and then issue a new WRITE command. Again, the WRITE command issues one padding character-time before sending the polling/address or beginning to send data. On reception of an EOB-LRC sequence, the 1050 acknowledges within less than one-character time. However, for safety, we will include a delay of one-character time.

Note: if we use the 3976 Modem Model 1 in half-duplex mode, the turnaround time is 150 ms. This means that three padding characters are sent instead of one.

## Addressing

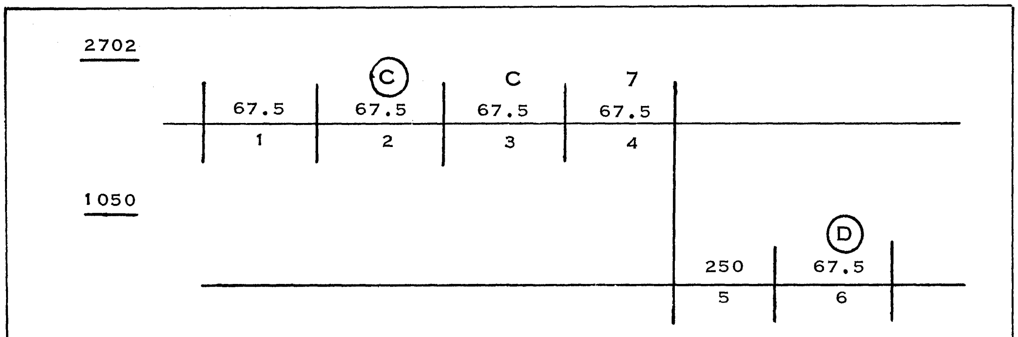
See Figure 73.



1. 67.5 MS = 1 PADDING CHARACTER
2. 67.5 MS = Ⓒ TO PLACE ALL TERMINALS IN CONTROL MODE
3. 67.5 MS = ADDRESS CHARACTER FOR TERMINAL A
4. 67.5 MS = COMPONENT SELECT CHARACTER
5. 250 MS = TERMINAL TURNAROUND TIME
6. 67.5 MS = POSITIVE ANSWER
7. 67.5 MS = QUIET TIME
8. 67.5 MS = PADDING CHARACTER
9. 67.5 MS = Ⓓ POSITIVE RESPONSE TO ADDRESSING

ADDRESSING TIME IS 790 MSEC.

Figure 73. Addressing: 2702 and 1050 Turnaround Time



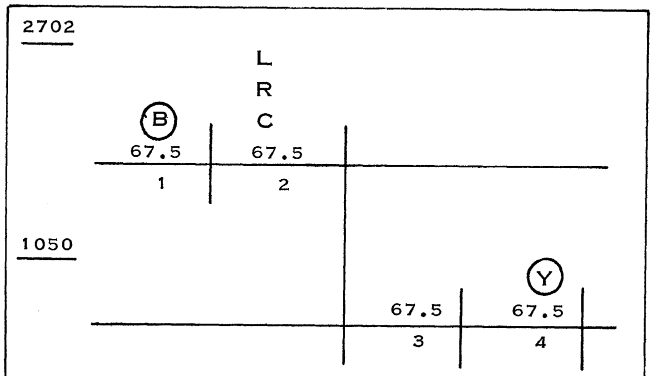
1. 67.5 MS = PADDING CHARACTER
2. 67.5 MS = Ⓒ TO PLACE ALL TERMINALS IN CONTROL MODE
3. 67.5 MS = ADDRESS CHARACTER FOR TERMINAL C
4. 67.5 MS = COMPONENT SELECT CHARACTER
5. 250 MS = TERMINAL TURNAROUND TIME
6. 67.5 MS = Ⓓ POSITIVE RESPONSE TO POLLING

POLLING TIME IS 588 MSEC.

Figure 74. Polling: 2702 and 1050 Turnaround Time

End of Block Conditions

See Figure 75.



1. 67.5 MS = B END OF BLOCK
2. 67.5 MS = LRC CHARACTER
3. 67.5 MS = TERMINAL TURNAROUND TIME
4. 67.5 MS = Y POSITIVE RESPONSE TIME

END OF BLOCK TIME IS 270 MSEC

Figure 75. End of Block: 2702 and 1050

Note: All the times shown in Figures 73, 74, and 75 exclude the channel response time due to, for example, byte transfer, command chaining, interrupts, etc., because this is dependent upon customer's programs and systems priorities.

B. Device Control Characters (Including Idle Characters)

There are different characters available to perform the necessary device functions which increase the line load by themselves. In addition, these device control characters also require idle characters to fill the time during execution of mechanical functions.

The influence on throughput of the different input and output components will be discussed in detail in the following paragraphs.

### 4. 2. 2 TIMING CONSIDERATIONS

First, we must examine in detail messages sent from the different input components to the computer.

Keyboard of the Typewriter

Timing for manual procedures, such as conversion from verbal or written material to the keyboard, is somewhat difficult to estimate. Fatigue factors and operator training discrepancies have to be taken into account. For example, even though operators are capable of keying as rapidly as eight strokes per second, the overall average rate will be close to 1.5 - 2 strokes per second. These figures account for carriage return, inserting forms, rest period, rekeying time for errors, etc.

If the messages are long, the rate could be somewhat higher. For known message lengths, the following calculating method can be used:

Split the operator-keyed characters as follows: the first character requires one second; and from 2 to 126 characters .5 seconds each. These times assume a well organized, readable document from which to key the message.

It is also advisable to include for safety two additional seconds for operator time per transaction. This time would be wasted by paper shuffling, operator response to polling, etc.

Note: When only numerical data must be keyed in, it is possible to have (on an RSDP basis) a numerical keyboard or an adding machine connected to the 1051. This can increase the operator rate to 3-5 strokes per second.

Card Reader

When we calculate effective speed, we need time for:

A. Reading: 67.5 x R
(R is the number of characters read from card and/or program tape.)

B. Feeding time to bring in the next card
954 msec. for 22-column cards
708 msec. for 51-column cards
462 msec. for 80-column cards

C. Card ejection time: 16.7 (S - L + 1)
16.7 msec. = eject speed per card column at 60 characters per second
S = size of card (80, 51, 22)
L = last column read
The 1 represents column 81 and is added for time to initiate feeding of the next card.

Note: With the line-correction special feature, the following additional timing considerations apply:

If the computed time is $\leq$ (less than or equal to) 205 msec., use 205 msec.

If the computed time is $>$ (greater than) 205 msec., use the calculated figure.

Without the line correction special feature, use the calculated time.

D. If the Card Reader Program Special Feature is used, and skipping occurs under program control, the rate is 67.5 msec per column, because each tape column must be read as skipping occurs.

When the High-Speed Skip Feature is installed, skipping is done at 16.7 msec per column.

To calculate the throughput, we have to add the amounts obtained in items A, B, C, and D above.

If there is a retransmission, we have to recirculate the card to column 1 so that it can be read again. This requires for a:

| | |
|---|---|
| 80-column card | 1,400 msec. |
| 51-column card | 1,900 msec. |
| 22-column card | 2,400 msec. |

Paper Tape Reader

As the paper tape is a continuous medium, no extra time is needed for a card reader. Additional time is necessary only in case of retransmission. The time we need for positioning the tape for reread is 67.5 msec. x C. (C is the number of characters in the tape message to be retransmitted.)

Calculations of the effective speed for sending messages from the computer to the different output units follow.

Typewriter

We must take care that as long as the typing element is moving to perform editing functions (carriage return, line feed, tab, etc.), no data characters are transmitted, since these characters would otherwise be lost. For this reason, we must provide, through programming, that enough idle characters are inserted to overlap the time to carry out the functions.

A. Carriage Return

The necessary time to perform this function depends on:

(1) length of the printed line,
(2) displacement speed of the typing element.

The CR function needs (1.5 + T) x 67.5 msec. (T is the distance in inches the typing element has to displace itself to come back to the margin. This time does not include the necessary time for the carriage-return character itself.)

With a speed of 14.8 cps, one character represents $\frac{1}{14.8}$ sec. = $\frac{1000}{14.8}$ msec. = 67.5 msec.

Therefore, the number of idle characters we have to insert to absorb the waiting time for CR is 1. 5 + T.

It is possible to order the accelerated carriage return, which will accomplish this function up to twice as fast as the normal speed. For instance, with this feature installed, the following timings apply:

| Length of CR in Inches | Number of Idle Char. Reg. |
|---|---|
| 0 - 2. 5 | 3 |
| 2. 5 - 5 | 4 |
| 5 - 7 | 5 |
| 7 - 10 | 6 |
| 10 - 13 | 7 |

B. Horizontal Tab

The necessary time to perform this function is (0. 75 + 1. 1 T) x 67. 5 msec. (T is the distance in inches the typing element displaces itself. This time does not include the 67. 5 msec. for the tab code which initiates the function. For the tab function, we have to insert 0. 75 + 1. 1 T idle characters. )

C. Line Feed

For this function, we need 2 x 67. 5 = 135 msec. If only line feed has to be performed, we have to insert two idle characters for every line feed.

D. Vertical Forms Control

This special feature provides controls for vertical movement of forms. This is controlled with two character codes (see Figure 76). All form movement occurs at line-feeding speed.

Two idle characters per line must be inserted. But only once per form movement do we need the prefix character in combination with the line feed or space character, instead of a line feed character for each line.

Note: With a 1050 system, it is possible to work simultaneously in a home- and in a line-loop operation. This means that, for example, we can receive the information in paper tape (line-loop) by means of the paper tape punch. The paper tape can be inserted in the reader and punched out in a home-loop operation. This is an advantage as far as line load is concerned, since no idle characters have to be inserted. In a home-loop operation, the reader is locked automatically during the time the printing element is moving. This is also valid if we read cards. Refer to Figures 77 and 78.

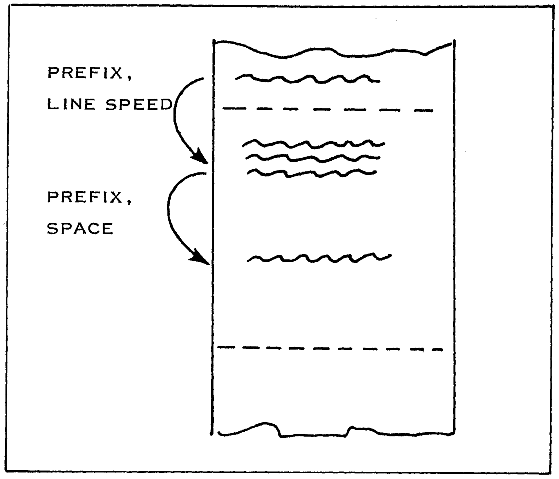


Figure 76. Vertical Forms Control

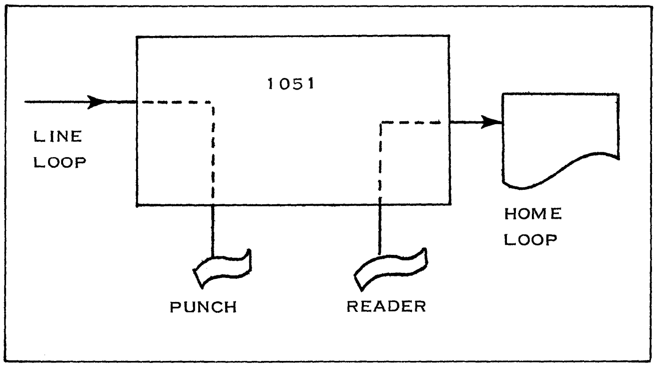


Figure 77. 1051 Control

Card Punch

When we calculate throughput for the cards to be punched, we need time for:

| | | |
|---|---|---|
| A. | Feeding and registering | 330 msec |
| B. | Punching | 14. 8 cps |

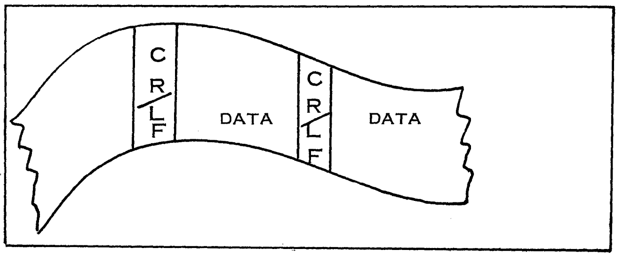


Figure 78. Home-Loop Operation

C. 1. Duplicating: 1057 - 20 cps (50 msec/column)
1058 - 10 cps (55 msec/column)

2. When duplication is started under program card control (an 0 or 6 punch with program card):

1057 75 + 50 n msec.
1058 75 + 55 n msec.
75 msec. = start time
n = number of columns duplicated

3. When duplication is started under computer control (a prefix character followed by an F character sent by the computer):

1057 165 + 50 n msec.
1058 165 + 55 n msec.
165 msec. = start time

D. Skipping or releasing with 80 cps:

85 + 12.5 n msec.
85 msec. = start time
12.5 msec. = one-character time at 80 cps
n = number of columns skipped or released.

If we receive information from the computer by a card punch, we must be very careful that, through programming, we insert enough idle characters to fill the time necessary to execute the different functions.

Paper Tape Punch

Here we are considering a continuous medium. Only if the Line Correction Special Feature is used must we take precautions.

When an incorrect message is received, a negative answer is delayed until the tape punch reverses the feeds to the previous CR/LF code and deletes forward until the eighth track punch is sensed. After this, the computer sends the message again. Between the time the message is first sent and the retransmission, enough idle characters have to be inserted to fill the time required to perform the functions described above. The time we need is:

(2 x 67.5) + 67.5 (2 D) msec.

D is the number of characters to be deleted (including EOB and CR/LF). The LRC time to complete the operation must be added.

### 4.2.3 CALCULATION EXAMPLES

Card Reading (Figures 79 and 80)

In this example, the throughput is calculated for sending punched cards from the 1050 system to the computer. The card reader is equipped with the card reader program and the High-Speed Skip Feature.

Data punched in the tape as constant data can also be read, thereby doubling card capacity (160 columns), if necessary. The tape and card move synchronously during the reading operation, a column of tape and a column of the card being read alternately, starting with column 1 of the tape. If data is punched in a tape column as well as in its corresponding card column, two reading cycles occur. If the tape is not punched, only one reading cycle occurs. For this calculation, we use the card model indicated in Figure 79. The skip start, skip stop punches, and the EOB code are punched in the program tape.

Reading of the cards is started by polling. This takes 588 ms. The total time we need to transmit n cards is 520 + n x 3,354 msec, as shown in Figure 80.

For a sufficiently large number, the time for polling may be disregarded. We can then say that

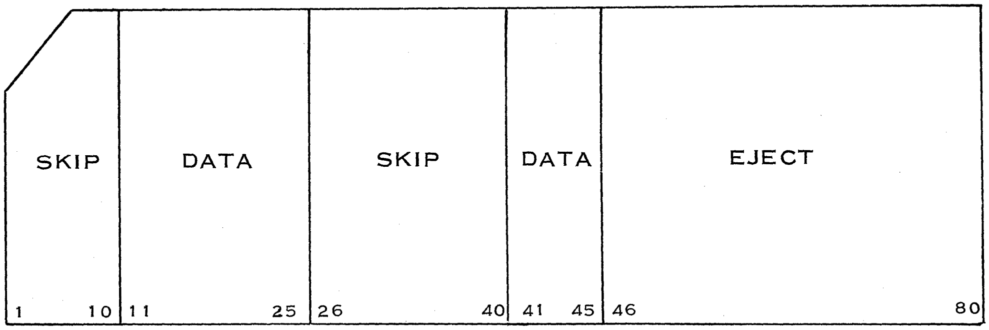


Figure 79. Card Model

| OPERATION | | | TIME REQUIRED |
|---|---|---|---|
| CODE | FUNCTION | CALCULATION | IN MSEC. |
| | FEEDING | – | 462 |
| SKIP START | | 1 X 67.5 | 67.5 |
| | SKIPPING | 10 X 16.7 | 167 |
| SKIP STOP | | 1 X 67.5 | 67.5 |
| DATA | | 15 X 67.5 | 1,012.5 |
| SKIP START | | 1 X 67.5 | 67.5 |
| | SKIPPING | 15 X 16.7 | 250.5 |
| SKIP STOP | | 1 X 67.5 | 67.5 |
| DATA | | 5 X 67.5 | 337.5 |
| EOB | LRC TIME | | 270.0 |
| | EJECT | 35 X 16.7 | 584.5 |
| | | TOTAL | 3,354.0 |

Figure 80. Card Operation: Timing

in 3,354 msec, 20 data characters are transmitted. This means an effective speed of $\frac{20}{3.36}$ = 6.0 characters per second. The format of the above cards (Figure 79) is very inefficient from the transmission point of view. Normally, we try to put different card fields next to each other in the cards.

From a detailed calculation (not using special features), we can derive the following figures for the effective speed of the card reader:

| | |
|---|---|
| Cards punched in columns 1-80 | 12 cps |
| Cards punched in columns 1-40 | 10 cps |
| Cards punched in columns 1-20 | 9 cps |

As a rule of thumb, we can say that the average effective speed will be about 10 cps.

This last figure can also be used for the card punch, paper tape punch, and paper tape reader.

Printing (Figures 81 and 82)

First example: For the first calculation example, the form is especially designed to reach a high effective speed, as shown in Figure 81.

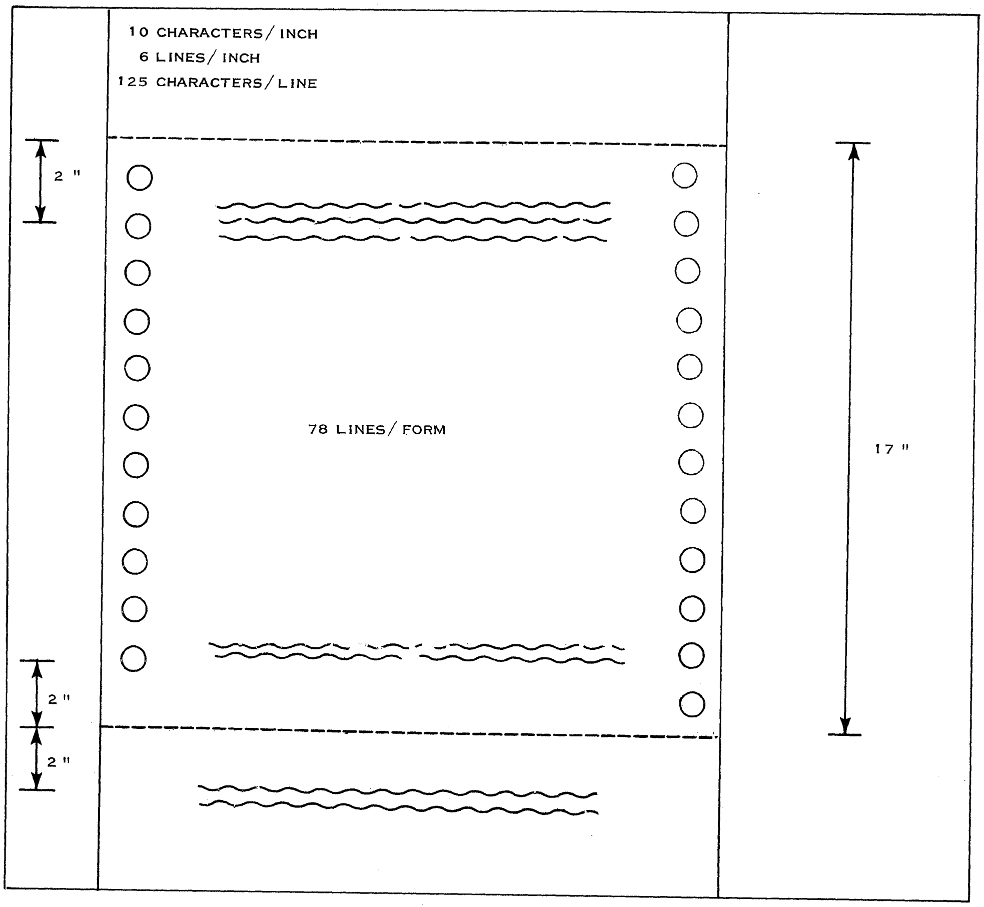


Figure 81. 78-Line Form

| OPERATION | | NO. OF | CALCULATIONS | TIME REQ. |
|---|---|---|---|---|
| CODE | FUNCTION | LINES | PER LINE | IN MSEC |
| CR/LF | | 78 | 1 X 67.5 | 5,265 |
| | CR/LF | 78 | (1.5 + 12.5) X 67.5 | 73,710 |
| DATA | | 78 | 12.5 X 67.5 | 658,125 |
| EOB | LRC-TIME | 78 | 270 | 21,060 |
| LF | | 23 | 1 X 67.5 | 1,552.5 |
| | LINE FEED | 23 | 2 X 67.5 | 3,105 |
| | | | TOTAL | 762,817.5<br>= 763 SEC. |

Figure 82. Printing Operation: Timings

Before the forms are printed, the typewriter is addressed. This takes 790 msec.

If a sufficiently large number of forms is printed within one addressing, we can ignore this value. On one form, 78 x 125 = 9,750 data characters are printed. This means that the effective speed is $\frac{9,750\ \text{cps}}{763}$ = 12.8 cps. The 763 seconds are the result of the example shown in Figure 82.

If there is a transmission error in one of the lines, a certain procedure can be organized by programming. For example:

1. Immediate retransmission of the line involved;

2. Go to a new form and repeat completely the contents of the last form;

3. Print out a special error message.

Second example: The next example is a less efficient layout for an inquiry procedure (see Figures 83 and 84).

After the terminal has been polled, the operator keys in the customer number with an assumed speed of 2 cps. This means $\frac{1000}{2}$ = 500 msec per character.

If we look only at the printing part (see Figure 84), we discover that 366 characters have been printed in

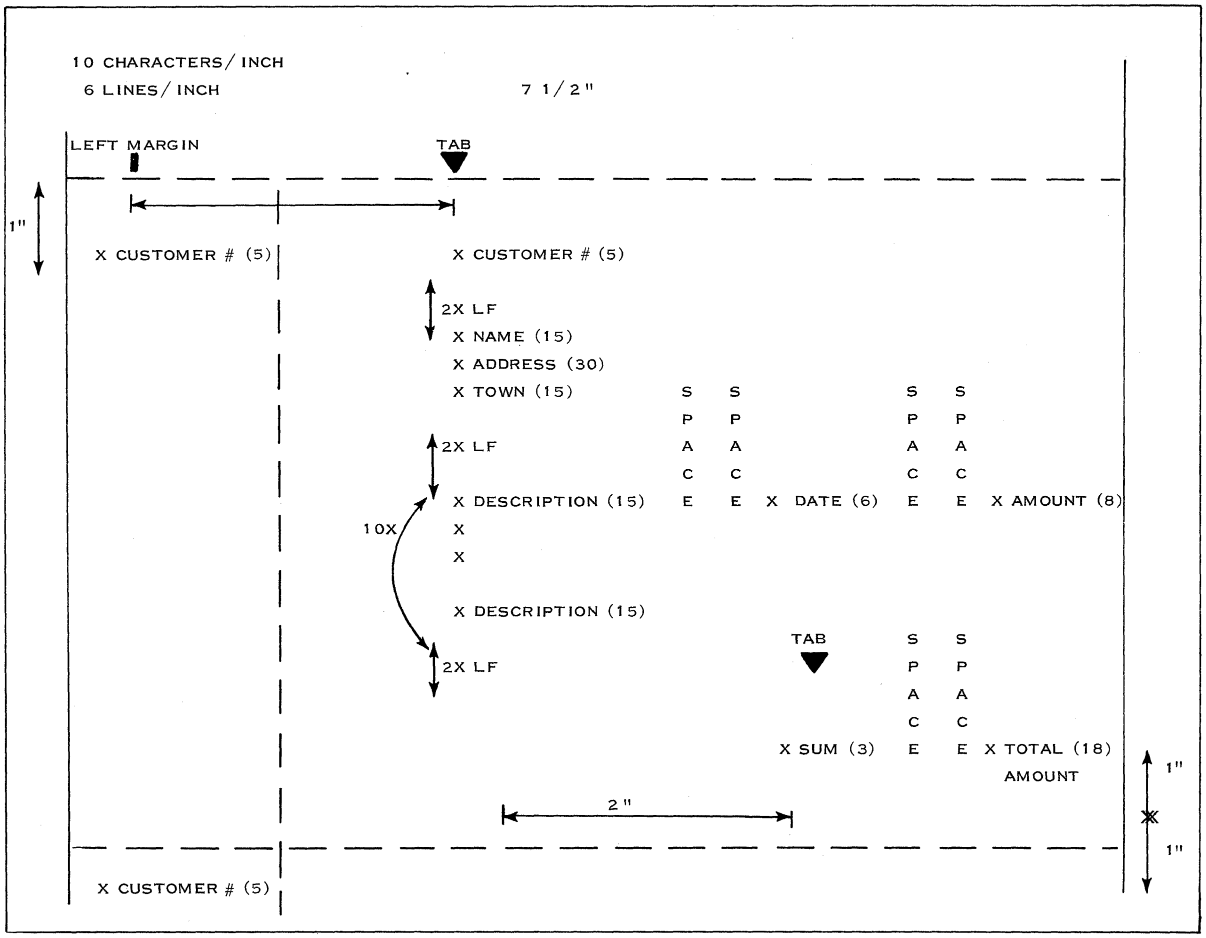


Figure 83. Layout for Inquiry Procedure

| OPERATION | | NO. OF | CALCULATIONS | TIME REQUIRED |
|---|---|---|---|---|
| CODE | FUNCTION | LINES | PER LINE | IN MILLISECONDS |
| | POLLING | | 588 | 588 |
| DATA | | | 5 X 500 | 2,500 |
| EOB – KEY | | | 1 X 500 | 500 |
| | LRC – TIME | | 270 | 270 |
| | | | TOTAL INPUT | 3,858 |
| | | | PROCESSING | |
| | ADDRESSING | | 790 | 790.0 |
| TAB | | | 1 X 67.5 | 67.2 |
| | TAB | | (0.75 + 1.1 X 2.5) X 67.5 | 236.3 |
| DATA | | | 5 X 67.5 | 337.5 |
| CR/LF | | | 1 X 67.5 | 67.5 |
| | CR/LF | | (1.5 + 3) X 67.5 | 303.8 |
| LF | | | 1 X 67.5 | 67.5 |
| | LF | | 2 X 67.5 | 135.0 |
| TAB | | | 1 X 67.5 | 67.5 |
| | TAB | | (0.75 + 1.1 X 2.5) X 67.5 | 236.3 |
| DATA | | | 15 X 67.5 | 1,012.5 |
| CR/LF | | | 1 X 67.5 | 67.5 |
| | CR/LF | | (1.5 + 4) X 67.5 | 371.3 |
| TAB | | | 1 X 67.5 | 67.5 |
| | TAB | | (0.75 + 1.1 X 2.5) X 67.5 | 236.3 |
| DATA | | | 15 X 67.5 | 1,012.5 |
| EOB | LRC – TIME | | 270 | 270.0 |
| LF | | | 1 X 67.5 | 67.5 |
| | LF | | 2 X 67.5 | 135.0 |
| CR/LF | | | 1 X 67.5 | 67.5 |
| | CR/LF | | (1.5 + 4) X 67.5 | 371.3 |
| TAB | | 10 | 1 X 67.5 | 675.0 |
| | TAB | 10 | (0.75 + 1.1 X 2.5) X 67.5 | 2,363.0 |
| DATA | | 10 | 33 X 67.5 | 22,275.0 |
| EOB | LRC – TIME | 10 | 270 | 2,700.0 |
| CR/LF | | 10 | 1 X 67.5 | 675.0 |
| | CR/LF | 10 | (1.5 + 5.8) X 67.5 | 4,927.5 |
| LF | | | 1 X 67.5 | 67.5 |
| | LF | | 2 X 67.5 | 135.0 |
| TAB | | | 1 X 67.5 | 67.5 |
| | TAB | | (0.75 + 1.1 X 2.5) X 67.5 | 236.3 |
| TAB | | | 1 X 67.5 | 67.5 |
| | TAB | | (0.75 +1.1 X 2) X 67.5 | 199.2 |
| DATA | | | 13 X 67.5 | 877.5 |
| EOB | LRC TIME | | 270 | 270.0 |
| CR/LF | | | 1 X 67.5 | 67.5 |
| | CR/LF | | (1.5 + 5.8) X 67.5 | 492.8 |
| LF | | 12 | 1 X 67.5 | 67.5 |
| | LF | 12 | 2 X 67.5 | 1,620.0 |
| | | | TOTAL OUTPUT | 46,759.5 |

Figure 84. Printing Timing Input and Output for Inquiry

46.8 seconds. In this case, the effective speed will be $\frac{366}{46.8} = 7.8$ cps for the output. In practice, we very seldom have such an efficient form layout as presented in the first example (Figures 82 and 83). However, we may expect situations such as presented in the second example. The conclusion would be that, with a reasonably efficient form layout, we can expect an effective speed of 10 cps; and with an inefficient layout, a speed of 8 cps.

These figures can be used as a rule of thumb.

### 4. 2. 4 DETERMINATION OF THE TERMINAL LOAD

From the information we receive from the customer or from actual measurements, we can derive the time interval during which we may expect the maximum number of transactions. This time interval may be one week per month, one day per week, one hour, half an hour, etc. This highest peak in traffic is called the "peak period".

The peak period must be defined by the customer's requirements. A certain number of transactions has to be processed. If, for example, the peak can be spread out over the whole day, the average load during that day is also the peak load.

To find the terminal load in percent, we multiply the volume figures during the peak period times by the time per transaction. Now, we have to divide the total time calculated above by the time period within which the transactions involved arrive. If, for example, during the peak period 30 transactions arrive within one hour and each transaction takes 50 seconds, the terminal load will be $\frac{30 \times 50}{3600} \times 100\% = 42\%$. When the 30 transactions arrive within half an hour, the terminal load will be $\frac{30 \times 50}{1800} \times 100\% = 84\%$. This result gives the terminal load figure for the system's present-day peak volumes. But we have to use the system's volumes at the time of installation, plus the expected growth during the period the system is to be used without change. Moreover, we have to consider the increase in volume realized merely because an effectively operating, fast, new system is available.

These considerations are explained by an example. For this, we take the inquiry application transaction calculated previously.

The result was that we need 3.8 seconds for input and 46.8 seconds per transaction for output. To fix the total time the terminal is occupied by this conversational transaction, we also have to include the internal response time. This is the interval between the time the last character is is keyed in and the first character of the response is printed out on the typewriter. We assume that this time will be 2 seconds. For safety, we add two additional seconds for operator time per transaction. This time is wasted by paper shuffling, operator response to polling, etc. If we summarize the times per inquiry transaction, we have:

| | | |
|---|---|---|
| Input | 3.8 | seconds |
| Processing time | 2.0 | " |
| Output | 46.8 | " |
| Operator time | 2.0 | " |
| Total | 54.6 | " |

We assume that during the peak hour there are 30 inquiries. This means that the total load is $\frac{30 \times 54.6}{3600} \times 100\% \simeq 46\%$.

We always have to expect some deviation from the average number of messages during a certain period. As the messages arrive at random, as shown in Figure 85, we cannot accept a 100% terminal load. With the help of the queuing theory, we can derive that if $\rho$ is the load of the terminal, the mean waiting time we have to accept before we can key in our transaction is $\frac{\rho}{1-\rho}$ the time per transaction. For example, if the terminal load is 75% and the transaction time is 54.4 seconds, we have to accept a mean waiting time of $\frac{0.75}{1 - 0.75} \times 54.4 = 163.2$ seconds. With a 60% terminal load, the mean waiting time will be $\frac{0.6}{1 - 0.6} \times 54.4 = 81.6$ seconds.

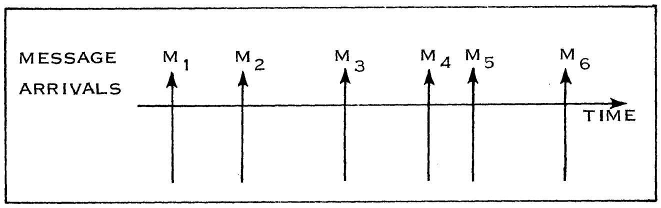


Figure 85. Example of Random Arrival of Messages

In general, it is recommended that a 75% terminal load for a point-to-point connection should not be exceeded.

For a multi-drop configuration, the line load is the addition of the load of the terminals connected to the same line. For a more detailed explanation of this subject, we refer to subsection 3. 3, Traffic Analysis and Network Design.

### 4. 2. 5 DIFFERENCES BETWEEN THE 2740 and 1050 TERMINALS

The 2740 is a typewriter-keyboard terminal with the same characteristics as the 1050. For example, the line control procedures and speed of the 1050 also apply to the 2740. When this terminal is equipped with the Station Control and Record Checking Features, it can be intermixed with 1050 terminals on the same line.

The Station Control Feature represents the addressing and polling capability of the 1050. The Record Checking Feature represents the LRC checking capability of the 1050.

Line Control for the 2740 with Record Checking and Without Station Control Feature (Figure 86)

In this case there is no polling or addressing. The operator starts to send a Ⓓ (End of Address) character, which is immediately followed by the message itself. A message sent from the computer to the terminal also starts with a Ⓓ character.

This technique can be used only when one terminal per line is used because in this case it is not necessary to determine which terminal is sending or receiving.

Line Control for the 2740 Without Record Checking and Without Station Control Features (Figure 87)

In this case, there is no polling or addressing. We can conclude that the number of line control characters involved in the transmission of a message is dependent on the availability of station control and/or record checking. However, for the calculation of the effective speed of the terminal, we can ignore this factor and use the same rules of thumb as developed for the 1050 system.

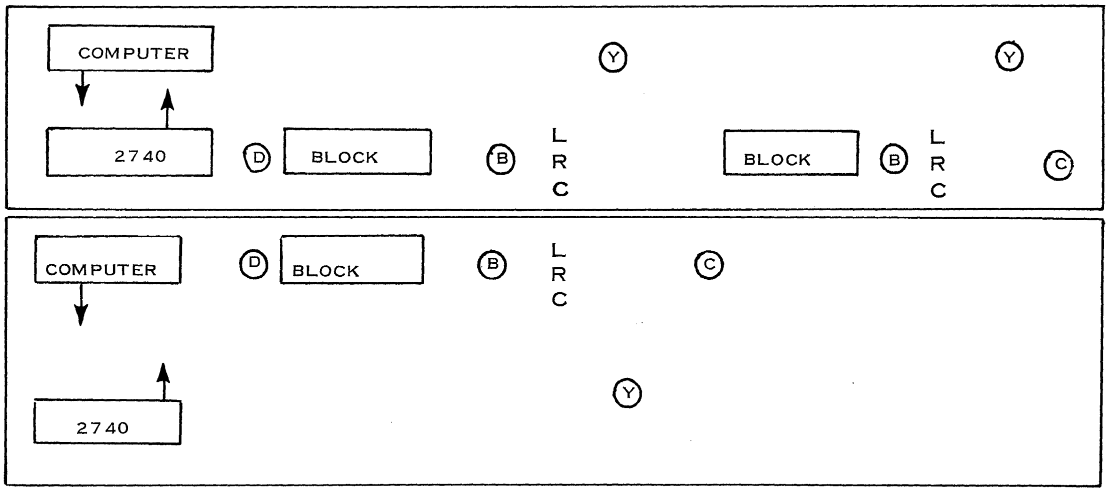


Figure 86. Line Control for the 2740 with Record Checking and Without Station Control Feature

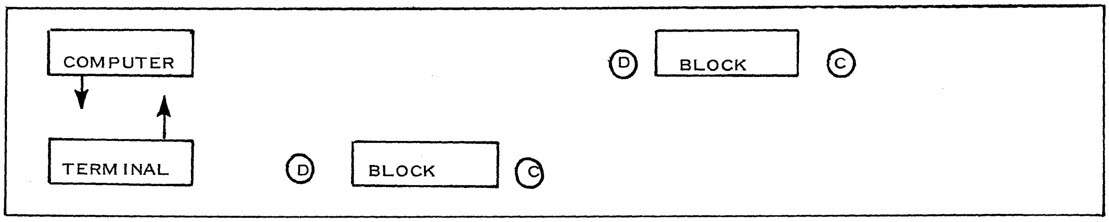


Figure 87. Line Control for the 2740 Without Record Checking and Without Station Control Feature

## 4. 3 TERMINAL SYSTEMS AND MULTIPLEXOR CONFIGURATION

### 4. 3. 1 TERMINAL SYSTEM CONFIGURATION IBM 1050 (Refer to Subsection 3. 6. 4)

In order to design a 1050 System, a configurator should be used.

All input and output components of a 1050 System are connected to a control unit 1051. There are three models:

1. 1051 Model 1 contains two separate data channels called "line" and "home" channel. Providing that enough components are connected to either of the channels, both can operate simultaneously. While, for instance, a paper tape reader transmits over the communication line, the operator can punch paper tape at the same time on the home channel, using the keyboard of the 1052 as an input component for the home channel.

2. 1051 Model 2 contains only the line channel. The data transmitted can be printed on a printer at the sending station at the same time.

3. 1051 Model N1 contains only a home channel. It can be used to produce cards or paper tape in offline operation. This might be necessary if, in remote locations, more paper tape must be punched than one 1051 System Model 1 can produce. In that case, only one additional 1051 Model N1 need be specified.

If a modem is not used to connect the 1050 System to a communication line, an IBM line adapter must be specified. Line adapters usually are used on private networks within a customer's installation or on leased lines.

IBM line adapters have certain specifications and restrictions. The SRL manual, Line Adapter for Data Communication, Form A24-3435, should be studied carefully before specifying an IBM line adapter. The shared line adapters can be used effectively to reduce line cost. If, in a remote location, more than one 1050 System is to be installed and the line load or response time requirements do not allow a multidrop operation on all 1050's on one leased line, this leased line can be used by several 1050 Systems (maximum of 4) simultaneously.

Every special feature of the configurator should be analyzed carefully, to ascertain whether it is necessary in a given application. For instance, if FORTRAN statements have to be transmitted or received by a 1056 card reader or 1057 card punch, respectively, the extended character reading or punching features must be specified. A printing card punch 1058 cannot be used to receive FORTRAN data because the feature is not available for a 1058.

### 4. 3. 2 MULTIPLEXOR CONFIGURATOR (Refer to Subsection 3. 6. 4)

A small communications-based system will normally use only the 2701 and 2702 multiplexors. The 2703, with a capacity to connect up to 176 communication lines, is therefore not mentioned here. The major hardware differences between the 2701 and 2702 lie in:

- the number of lines that can be connected,
- the method of using line adapters.

A configurator should be preapred for the multiplexors

2701 Data Adapter Unit

The 2701 can connect a maximum of 4 lines, depending on the terminal type. The adapters for the different terminal types are divided into different categories. The 1050 System belongs in Category 1. Up to 4 lines can be connected to one 2701. For the second, third, and fourth line an expansion feature is also necessary besides the terminal adapter. The third line also requires the expanded capability.

If modems are not used to connect the 1050 lines to the 2701, line adapters must be specified. There is only one line adapter type available on the 2701 to connect 1050 lines. This is the limited distance (13 km, or 8 miles) Type II line adapter. This line adapter is compatible with the Type II line adapters for limited distance (13 km, or 8 miles) in the 1050 System.

2702 Transmission Control Unit

The 2702 Transmission Control Unit can connect a maximum of 31 lines with 1050 systems. The basic unit is 15 lines, which can be expanded by another 16 lines. If modems are used, each line must have a data set line adapter to connect to the modem. If no modems are used on privately owned networks or on leased lines, IBM line adapters can be specified. IBM line adapters can be used in two different ways: Up to 16 IBM line adapters can be housed in the 2702 itself; if more lines are necessary, up to 30 lines can be connected with IBM line adapters which are housed in separate units, the 2711. In the 2702, in this case, for each line adapter of the 2711, a data set line adapter must be specified. One additional IBM line adapter, the 31, can be housed directly in the 2702.

## 4. 4 SIMPLIFIED PROCEDURE FOR THE DESIGN OF A CBS WITH 1050-2740

### 4. 4. 1 INTRODUCTION (Refer to Subsection 3. 12)

We will consider here only a system that works in conversational mode with 1050 or 2740 typewriter-keyboard terminals, one terminal per line.

A simplified system design procedure is shown for such systems.

This might be helpful because frequently any of the following situations may exist.

1. The proposal must be prepared on short notice.
2. Only average traffic figures are available.
3. The ideas concerning the message formats and the form layouts are very vague.
4. Until now the application to be put online is done on a batch basis.
5. There are no special response time requirements.
6. The system will consist of up to 6 lines used with one terminal per line.

| DESIGN OF | IS DEPENDENT ON | ANALYZE |
|---|---|---|
| CPU | | |
| • MODEL (PROCESSING SPEED) | • PROCESSING TIME | NUMBER OF INSTRUCTIONS |
| | • INTERFERENCE | CHANNEL LOAD |
| • CORE SIZE | • SOFTWARE SUPPORT | MESSAGE TRAFFIC, INPUT/OUTPUT |
| | • BUFFERS | MESSAGE TRAFFIC AND FORMATS |
| | • QUEUES IN CORE | TRAFFIC AND PROGRAM LOGIC |
| | • PROGRAMS | APPLICATIONS |
| INPUT/OUTPUT | | |
| • TYPE AND NUMBER OF TERMINALS | • INPUT/OUTPUT MEDIA AND AMOUNT | TRAFFIC AND APPLICATION TYPE |
| • TYPE OF TRANSMISSION CONTROL UNIT | • NUMBER OF LINES | NUMBER OF TERMINALS PER LINE |
| • DISK CAPACITY | • SIZE OF FILE | NUMBER OF RECORDS |
| | • MESSAGES TO LOG | MESSAGE TRAFFIC |
| | • QUEUES ON DISK | MESSAGE TRAFFIC |
| • TYPE AND NUMBER OF DISK FILES | • DISK UTILIZATION | MESSAGE TRAFFIC AND PROGRAMS |
| CHANNELS | • LOAD | MESSAGE TRAFFIC AND FILE ACCESSES |

Figure 88. Factors in System Design

Figure 88 shows the parts that must be designed, the factors they depend on, and what we have to analyze.

We see that to design the CPU, we need to analyze the overall input/output for the different devices and the channels. To design the channels we have to analyze the input/output for the different devices.

We have the following comments concerning Figure 88:

It has already been proved that the contribution to channel loading and CPU interference caused by the addition of a small CBS application to batch processing is very low and can be ignored.

Concerning disk utilization, we can say that if we remain within four accesses per second for a 2311 disk unit, utilization will be less than 60%.

As with a small CBS application we normally remain far beyond a 60% loading of the channel's disk units and the CPU, we can ignore the effect of queuing within the computer.

The conclusion is that with the above procedures we can confine ourselves to determining:

1. Internal response time;

2. Type, configuration, and number of terminals.

3. Type and configuration of transmission control unit;

4. Necessary disk capacity;

5. Core size of CPU.

### 4. 4. 2 INTERNAL RESPONSE TIME

Internal response time of the computer system is determined mainly by:

Number of file accesses per second:
If the number of accesses is equal to or less than two accesses per second (which is usual in a small CBS application), 170 msec for every read or write can be used as a rule of thumb for a 2311 or 2314. This also applies to the accesses necessary to read in a program or program part. This includes the BDAM macro time $\leq$ 20 ms.

#### Message Control Program

A conservative estimate of CPU time necessary for the message control program using BTAM can be obtained from Figure 89.

| MESSAGE LENGTH = $\frac{IN + OUT}{2}$ | MODEL 30 | MODEL 40 | MODEL 50 |
|---|---|---|---|
| 10 CHARACTERS | 11 MS | 6 MS | 4 MS |
| 40 CHARACTERS | 44 MS | 24 MS | 16 MS |
| 100 CHARACTERS | 70 MS | 40 MS | 24 MS |
| 200 CHARACTERS | 140 MS | 80 MS | 48 MS |
| 300 CHARACTERS | 210 MS | 120 MS | 72 MS |
| 400 CHARACTERS | 280 MS | 160 MS | 96 MS |

Figure 89. Timings for Message and Line Control Program Excluding BTAM READ and WRITE

In these timings are included the sending and receiving of messages, error control routines, and checking and analyzing of the message itself and the header information. For BTAM timings, refer to subsection 3. 12, Figure 58.

#### Application Program

For the application program, we must estimate the number of instructions. This number is multiplied by the average instruction speed for the System/360 involved (Figure 90).

### 4. 4. 3 TYPE OF CONFIGURATION AND NUMBER OF TERMINALS

Type and configuration of terminals we choose depends mainly on:

1. The number of messages to be handled during the peak period.

2. Point-to-point or multi-drop network (with or without buffering facilities)

| /360 MODEL | AVERAGE INSTRUCTION SPEED IN MICROSECONDS |
|---|---|
| 30 | 35 |
| 40 | 20 |
| 50 | 12 |

Figure 90. Application Program: Instruction Speed

3. Environment of operations
(i. e., factory, sales office, bank, etc.)

4. How the information is available
(documents, cards, paper tape).

For the simplified procedure, we limit ourselves to the typewriter-keyboard terminals (1050 - 1052 or 2740) working in conversational mode with the computer.

To calculate the effective speed, we use the following figures:

Input:
The average operator speed with a typewriter keyboard will be about 2 characters/second. This means $\frac{1000}{10}$ = 100 ms/character. With an inefficient form layout the speed is about 8 characters/second. (This means 125 ms/character.)

The time that the terminal is occupied per transaction is composed as follows:

| | |
|---|---|
| number of characters keyed in x 500 ms | ---- ms |
| operator time | 2,000 ms |
| internal response time | ---- ms |
| (number of printed characters) x 100 or 125 ms. | |
| Total time per transaction | ---- ms |

As the internal response time of the computer normally remains within 1 to 2 seconds (which is very short in comparison with the transmission times per transaction), it is also possible to disregard the internal response time for calculation of the terminal load.

Now, the first step is to calculate how long the terminal is occupied by the different types of transactions during the peak period.

A good period for which to design is the peak hour of the average day of the peak month, assuming that the degradation during the peak could be tolerated. If the amount of traffic during the peak period is unknown, we must at least consider that the traffic during the peak hour is twice that of an average hour.

The terminal load is calculated as follows:

$$\frac{\text{Time terminal is busy during peak period}}{\text{peak period}} \times 100\%$$

It is recommended that this figure should not exceed 75% per terminal.

Moreover, from the reliability point of view, it is recommended not to exceed 1.5 to 2 million printed characters per month, per typewriter.

Note: Remember that you will be installing this system about 10 to 24 months from now and the volumes will grow during this period, plus the fact that this system should be able to handle the job for a number of years to come. Also consider the increase in volume realized merely because the system is available.

### 4. 4. 4 TYPE AND CONFIGURATION OF TRANSMISSION CONTROL UNIT

The type of transmission control unit required is determined by the number of lines and the type of terminals to be connected.

In general, we can state:

| | | |
|---|---|---|
| Speed | > 600 bps | 2701 |
| Speed | < 600 bps | 2701, 2702, 2703 |

In the latter case, the criterion to choose is a matter of minimum price: for example, for the 1050 and 2740 terminals:

| | | |
|---|---|---|
| - | 4 lines | 2701 |
| - | Between 5 and 31 lines | 2702 |
| - | 32 lines | 2703 |

To configurate these units, we have to choose the necessary adapters for the terminals we are going to use. This can be done with the Sales Manual and the available configurators in this Handbook.

### 4. 4. 5 NECESSARY DISK CAPACITY

To calculate, refer to the subsection on Disk Capacity, 3. 8, which provides a simple method.

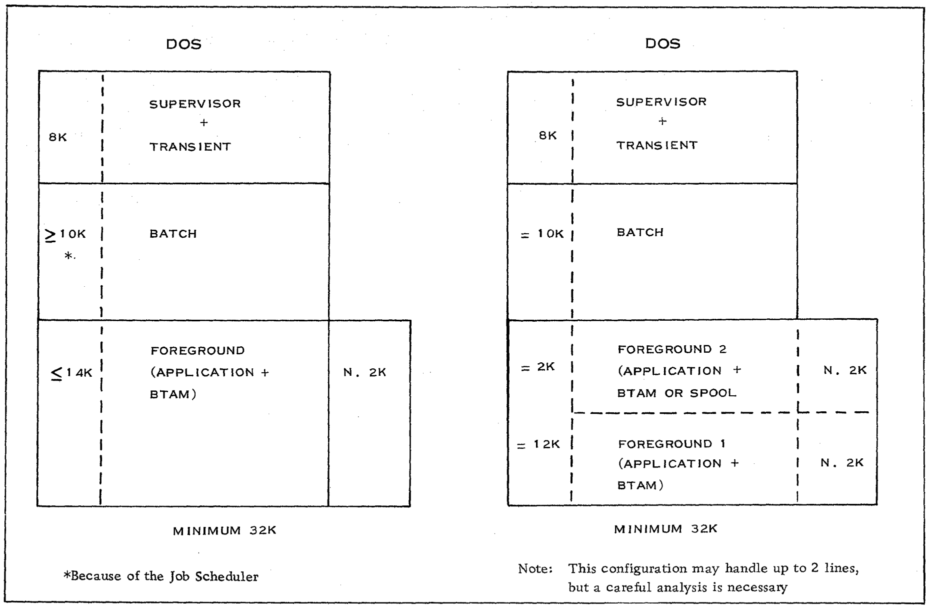
### 4. 4. 6 CORE SIZE OF THE CPU (Refer to Subsection 3. 7)

We can split up the CPU in the following principal parts:

Control program (DOS or SPS)
Message control program (BTAM)
Customer's application programs
Error recovery routines
Input/output areas (buffers)
Queues in core

Control Program

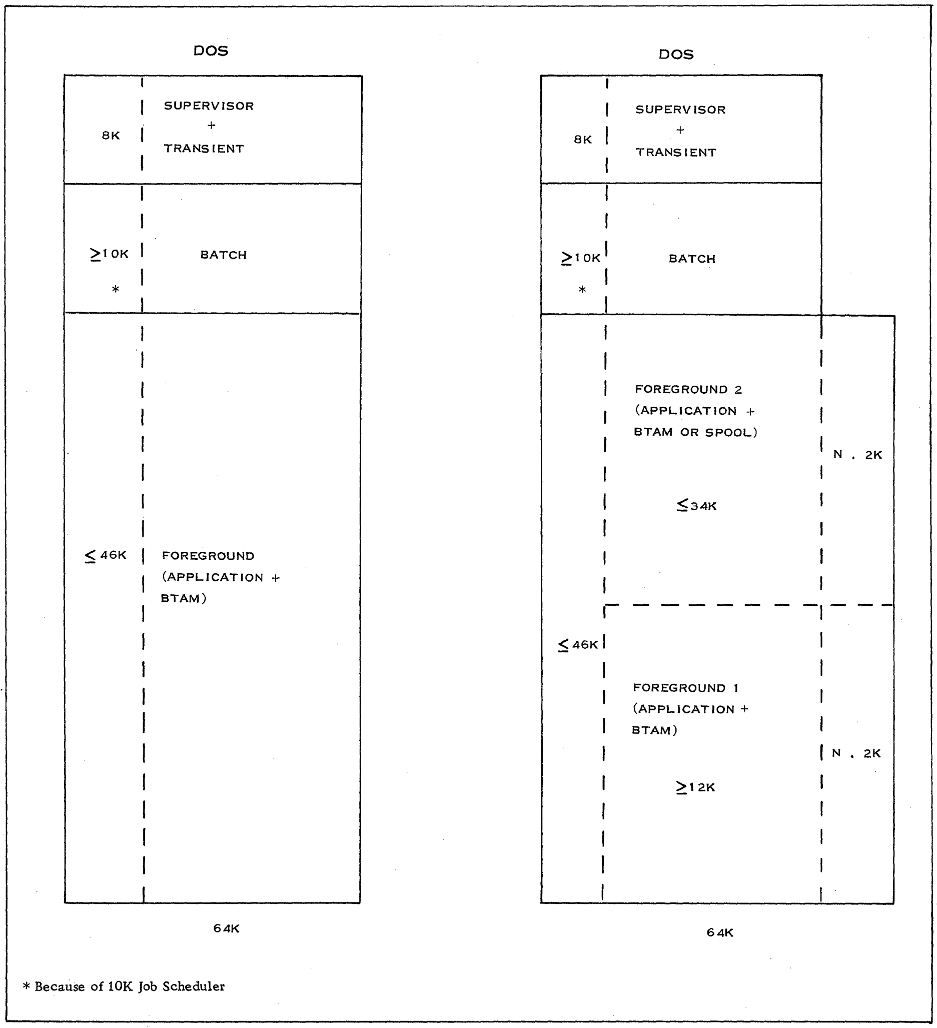
Figures 91, 92, and 93 show some possible divisions of the CPU for DOS and SPS in a System/360.



*Because of the Job Scheduler

Note: This configuration may handle up to 2 lines, but a careful analysis is necessary

Figure 91. Segments of the CPU for DOS for 32K



* Because of 10K Job Scheduler

Figure 92. Segments of the CPU for DOS, 64K

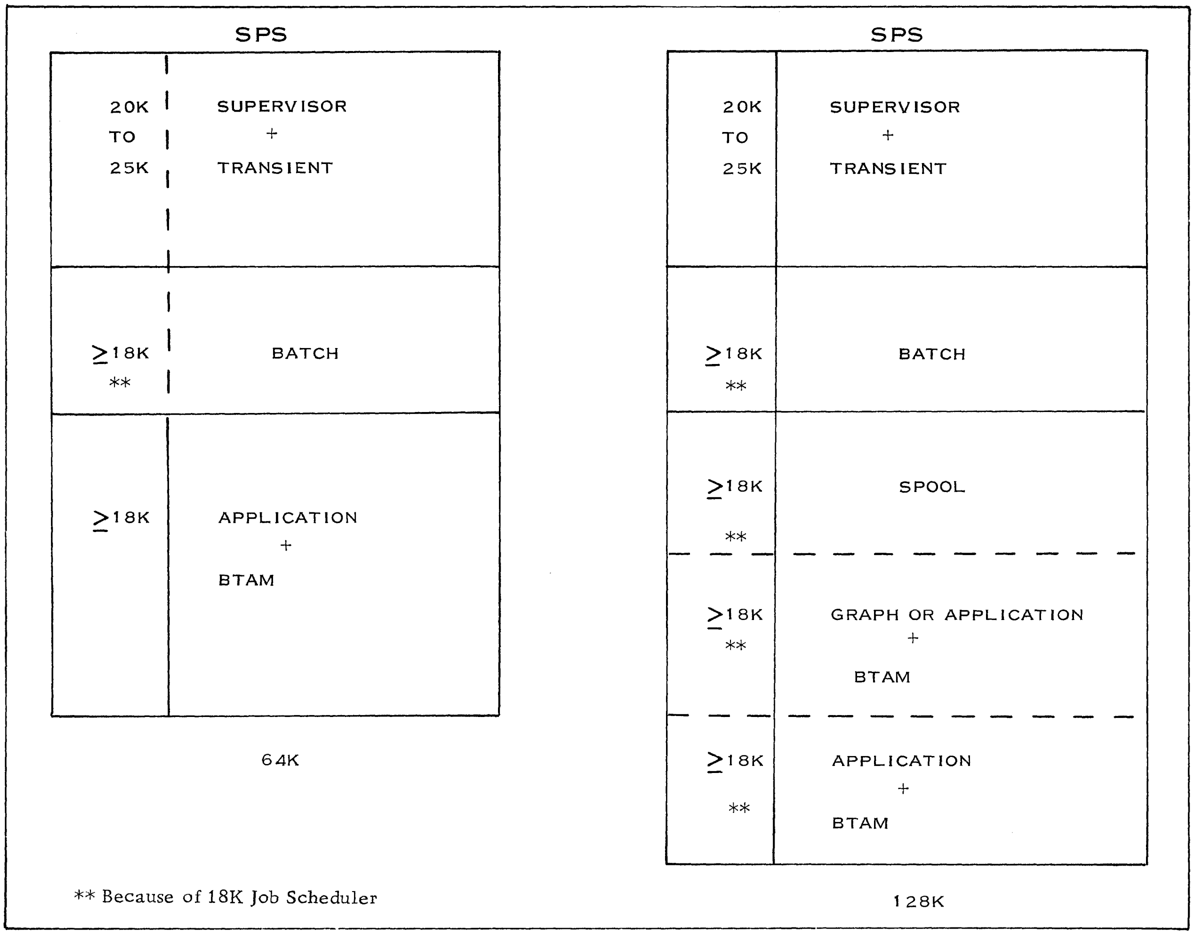


Figure 93. Segments of the CPU for SPS, 64K and 128K

| TYPE OF TERMINAL USED | NUMBER OF BYTES |
|---|---|
| 1050 OR 2740 | 2850 N. 260 |
| 1050 AND 2740 (WITH STATION CONTROL AND RECORD CHECKING) | 2850 N. 260 |
| 1050 AND 2740 (WITHOUT STATION CONTROL AND/OR RECORD CHECKING) | 3000 N. 260 |
| | n = number of terminals |

Figure 94. BTAM Core Requirements with Fixed 125-Byte Buffer Per Line and One Terminal Per Line, Including Communication Serviceability Features

As the speed of the 1050-2740 terminals is 14 cps (67.5 msec/character), one buffer per line is sufficient. The time to fill the buffer from disk after it is emptied (for example, in case of segmenting the message) has hardly any bearing on the effective speed. As the length of a printing line normally does not exceed about 100 characters, we use a 125-byte buffer per line as a rule of thumb. The idle characters necessary to execute the carriage return are also included in this.

The buffer capacity mentioned above is included in the core requirements for BTAM, as shown in Figure 94.

### 4. 4. 7. EXAMPLE

We will apply the simplified system design procedure to the 1050 example described in subsection 4. 5. As explained in the introduction, the only limitation is that we do not consider the multi-drop line. We have just presented the basic figures we need for the design. This information is derived from the flowcharts for the different type of transactions given in subsection 4. 5. For "rush order" transactions, we can disregard the effect of the branches on "minimum inventory master warehouse" and "minimum inventory zone warehouse". This can be derived from the detailed considerations given in the 1050 example. In Figure 95, we indicate the subjects which are considered in the different tables.

The figures that follow show the simplified design procedure as applied to the example given in subsection 4. 5. The figures indicate the following:

96. Internal Response Time
97. Terminal Load Per Transaction
98. Evaluation of Number of accesses Per Second
99. Total Terminal Load
100. Traffic in Characters Per Month by Terminal

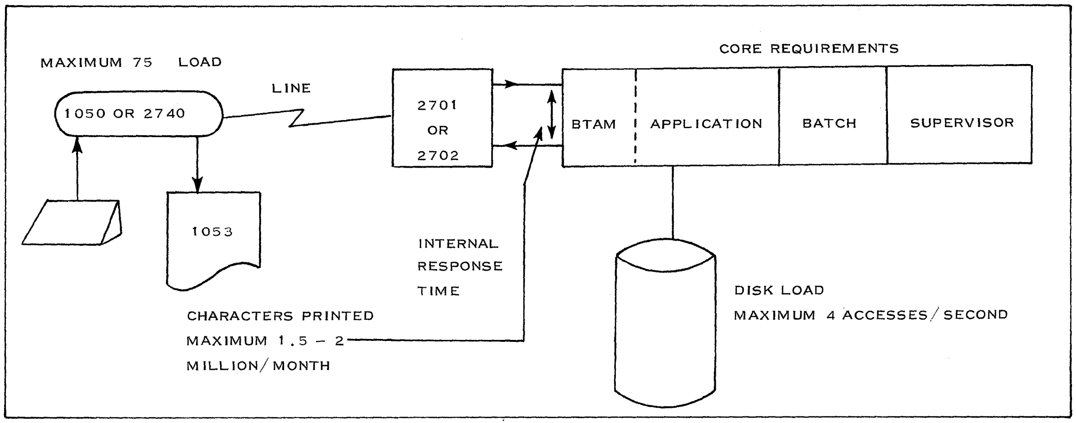


Figure 95. Example of Simplified Design: 1050 or 2740 CBS

| TYPE OF TRANSACTION | INPUT/ OUTPUT | DISK | | INSTRUCTIONS | | MESSAGE CONTROL BTAM | BTAM READ + WRITE | INTERNAL RESPONSE TIME |
|---|---|---|---|---|---|---|---|---|
| | | ACCESS TRANSACTIONS | RESPONSE TIME | NUMBER | PROCESSING TIME | | | |
| INQUIRY | I +O | 1 | 170 MS | 500 MS | 18 MS | 17 MS | 14 MS | 219 MS |
| STOCK UPDATING | I + O | 4 | 680 MS | 750 MS | 27 MS | 22 MS | 14 MS | 743 MS |
| RUSH ORDERS | I + O | 10 | 1,700 MS | 1,500 MS | 53 MS | 81 MS | 14 MS | 1,848 MS |

Figure 96. Internal Response Time

| TYPE OF TRANSACTION | NO. OF CHARACTERS | MEDIA | | EFFECT. SPEED IN CPS | TIME CHARACTER | OPERATOR TIME | TOTAL TIME | INTERNAL RESPONSE TIME | TERMINAL LOAD PER TRANSACTION |
|---|---|---|---|---|---|---|---|---|---|
| | | INPUT | OUTPUT | | MS | MS | MS | MS | MS |
| INQUIRY | 6 | KEYBOARD | | 2 | 500 | 2,000 | 5,000 | | |
| | 25 | | PRINTER | 10 | 100 | | 2,500 | | |
| | 31 | | | | | | | 219 | 7,719 |
| STOCK | 10 | KEYBOARD | | 2 | 500 | 2,000 | 7,000 | | |
| UPDATING | 35 | | PRINTER | 10 | 100 | | 3,500 | | |
| | 45 | | | | | | | 743 | 243 |
| RUSH | 29 | KEYBOARD | | 2 | 500 | 2,000 | 16,500 | | |
| ORDERS | 206 | | PRINTER | 8 | 125 | | 25,750 | 1,848 | 44,098 |
| | 235 | | | | | | | | |
| MINIMUM INVENTORY | 33 | | PRINTER | 10 | 100 | | 3,500 | | |

Figure 97. Terminal Load Per Transaction

| TYPE OF TRANSACTION | NO. OF TRANSACTIONS PER PEAK HOUR | ACCESSES PER TRANSACTION | NO. OF ACCESSES PER HOUR | NO. OF ACCESSES PER SECOND |
|---|---|---|---|---|
| INQUIRY | 167 | 1 | 167 | |
| STOCK UPDATING | 33 | 4 | 132 | $\frac{2529}{3600} =$ |
| RUSH ORDERS | 223 | 10 | 2,230 | |
| | | | 2,529 | 0.7 ACCESSES/ SECOND |

Figure 98. Evaluation of Number of Accesses Per Second

| CITY | TYPE OF MESSAGE | NO. OF MESSAGES PER PEAK PERIOD<br>– A – | TIME PER TRANSACTION IN SECONDS<br>– B – | A X B | PEAK PERIOD IN SECONDS | TERMINAL LOAD<br>$\frac{\sum A \times B}{C}$ |
|---|---|---|---|---|---|---|
| A | INQUIRY | 39 | 7.7 | 300.3 | | |
| | STOCK UPDATING | 6 | 11.2 | 67.2 | | |
| | RUSH ORDERS | 42 | 44.1 | 1,843.8 | | $\frac{2,231.1}{3,600}$ |
| | MINIMUM INVENTORY | 6 | 3.5 | 19.8 | | |
| | | | | 2,231.1 | 3,600 | 62 % |
| B | INQUIRY | 36 | 7.7 | 277.2 | | |
| | STOCK UPDATING | 6 | 11.2 | 67.2 | | $\frac{2,319.9}{3,600}$ |
| | RUSH ORDERS | 45 | 44.1 | 1,975.5 | | |
| | | | | 2,319.9 | 3,600 | 65 % |
| C | INQUIRY | 30 | 7.7 | 231.0 | | |
| | STOCK UPDATING | 9 | 11.2 | 100.8 | | $\frac{2,570.7}{3,600}$ |
| | RUSH ORDERS | 51 | 44.1 | 2,238.9 | | |
| | | | | 2,570.7 | 3,600 | 72 % |
| D | INQUIRY | 36 | 7.7 | 277.2 | | |
| | STOCK UPDATING | 6 | 11.2 | 67.2 | | $\frac{2,319.9}{3,600}$ |
| | RUSH ORDERS | 45 | 44.1 | 1,975.5 | | |
| | | | | 2,319.9 | 3,600 | 65 % |

Figure 99. Total Terminal Load

| CITY | TYPE OF TRANSACTION | MESSAGES PER DAY | WORKING DAYS PER MONTH | MESSAGES PER MONTH | PRINTED CHARS. PER MESSAGE | PRINTED CHARS. PER MONTH |
|---|---|---|---|---|---|---|
| A | INQUIRY | 100 | 20 | 2,000 | 31 | 62,000 |
| | STOCK UPDATING | 10 | 20 | 200 | 45 | 9,000 |
| | RUSH ORDERS | 110 | 20 | 2,200 | 235 | 517,000 |
| | MIN. INVENTORY | 15 | 20 | 300 | 33 | 6,600 |
| | | | | | | 594,600 |
| B | INQUIRY | 90 | 20 | 1,800 | 31 | 55,800 |
| | STOCK UPDATING | 15 | 20 | 300 | 45 | 13,500 |
| | RUSH ORDERS | 120 | 20 | 2,400 | 235 | 564,000 |
| | | | | | | 633,300 |
| C | INQUIRY | 80 | 20 | 1,600 | 31 | 49,600 |
| | STOCK UPDATING | 20 | 20 | 400 | 45 | 18,000 |
| | RUSH ORDERS | 130 | 20 | 2,600 | 235 | 611,000 |
| | | | | | | 678,600 |
| D | INQUIRY | 90 | 20 | 1,800 | 31 | 55,800 |
| | STOCK UPDATING | 15 | 20 | 300 | 45 | 12,500 |
| | RUSH ORDERS | 120 | 20 | 2,400 | 235 | 564,000 |
| | | | | | | 633,300 |

Figure 100. Traffic in Characters/Month by Terminal

## Core Requirements

In this system, we have considered:

1 line with a 1050 terminal,

3 lines with a 2740 terminal (without station control and record checking feature)

From the table on BTAM core requirements (Figure 94), we can derive that we need:

3000 + n x 260 + 2500 = 5500 + 4 x 260 = 6540 bytes

We must determine the core necessary for the application program itself in the same way as we do for the batch programs. Normally, we would include an overall safety factor of 20%.

## 4. 5 EXAMPLE: 1050 SYSTEM

### 4. 5. 1 GENERAL COMMENTS ON SPECIFICATIONS FOR THE INQUIRY AND UPDATING SYSTEM

These specifications are for a customer who has a System/360, Model 30, performing batch-processing with DOS, or for a customer who does not have an existing system but requires a batch-processing system with some communications capability. In both cases, the customer does not have an existing network.

The size of the communications network will be five lines, operating half-duplex up to a speed of 200 bps.

The system will be designed with 1050 and 2740 terminals.

The type of lines considered are:

- point-to-point (in-house)
- point-to-point (leased)
- multi-drop (leased)

The application described in the specifications is for a sales order entry system. This involves entering and processing of orders as well as updating items, customer records and inventory inquiries.

Other applications may fall under the same general procedure, such as reservations, banking and insurance systems.

The real-time portion of the application can start as a simple inquiry system. Inquiry is a function that can be applied without exception to any application.

Basically, the inquiry for a real-time system operates in the following manner:

1. The inquirer wants to know the current status of a certain bit of information.
2. He sends a formatted inquiry message from a terminal to a central computer.
3. The computer recognizes the inquiry type.
4. The computer than goes to the proper files and obtains the updated information.
5. The information is placed in a proper format by the computer.
6. The answer to the inquiry is sent back to the terminal.

The actual updating of the item records is done at the time a message is received from a terminal.

The final step is to put the actual sales order entry (only rush orders) application program online. However, customer records will be updated later in a batch-processing program.

### 4. 5. 2 CASE PROBLEM DESCRIPTION

A. General System Definition

1. This system is to be a real-time system for inquiry, updating item records when merchandise comes in, processing rush orders, and invoicing them immediately. Moreover, minimum inventory indication messages concerning the master warehouse are printed on the in-house terminal. Minimum inventory-indication messages concerning the zone warehouse are stored on disk.
2. When there is no real-time operation, the system processes mail or hand-delivered orders on a batch basis. The information is read from cards via a high-speed local card reader and the output is placed in printed form on a local high-speed printer. Furthermore, updating and accounting of customer records is done during this time on a batch basis.
3. The real-time operation is performed eight hours per day from 8:00 - 12:00 A. M. and 2:00 - 6:00 P. M.

B. Organization

The "problem" system includes seven zone warehouses, one of which also serves as a master warehouse. The master warehouse is the location of the computer center. Each of the zone warehouses has a remote terminal.

Customers ordinarily order by mail from the master warehouse. These items would be sent direct from the master warehouse to the customer. Replenishment of the zone warehouse is via the master.

There is a customer service area at each of the zone warehouses to answer customer inquiries and to execute rush orders.

C. Relation of the Zone Warehouses to the Master Warehouse

Figure 101 shows the relation of the zone warehouses

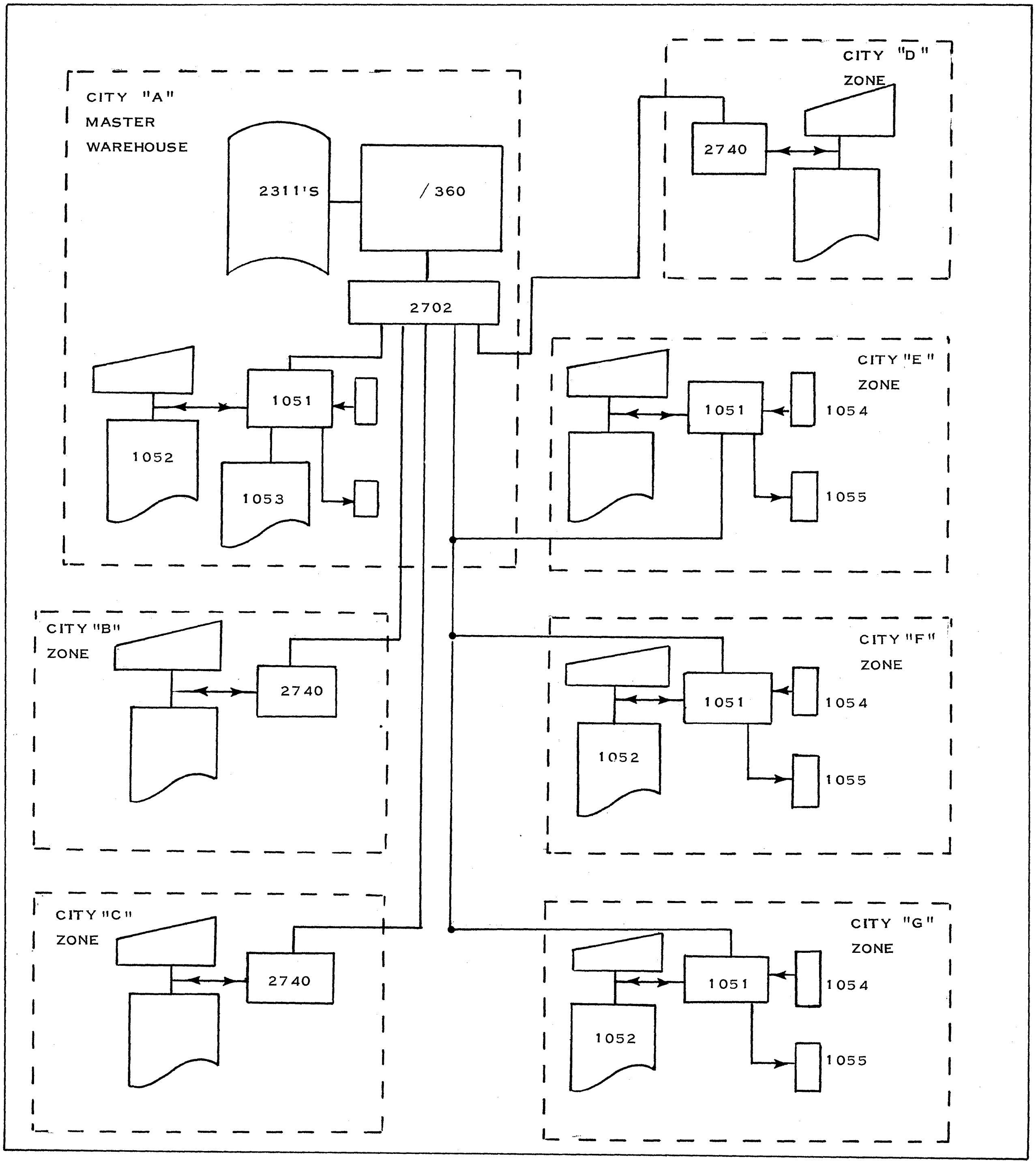


Figure 101. Relation of Zone Warehouses to Master Warehouses

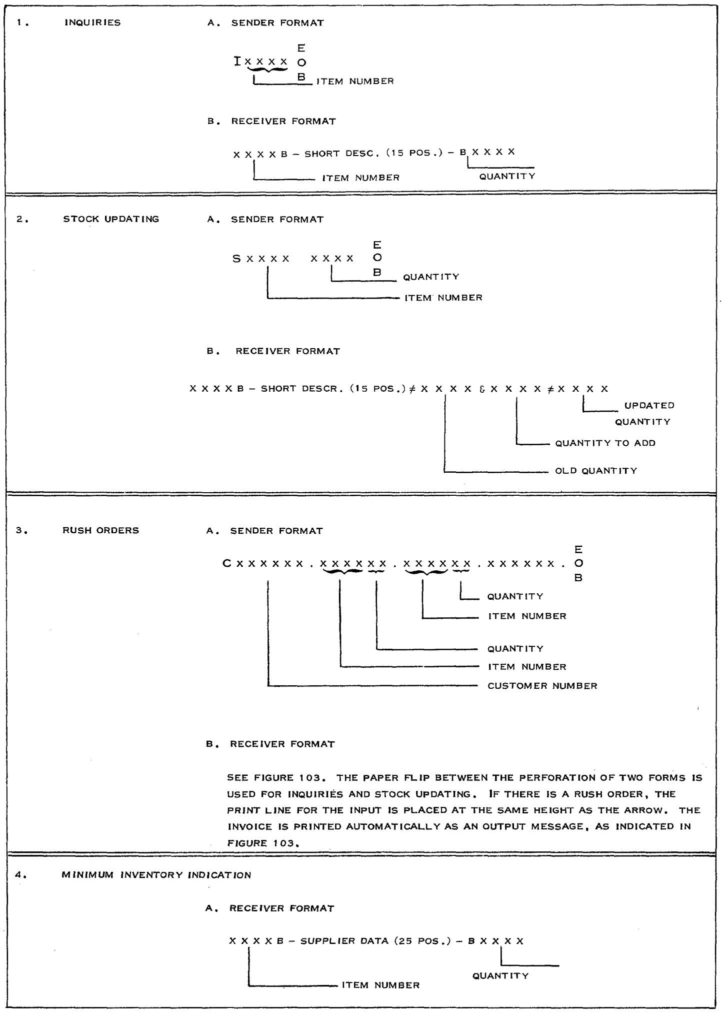
to the master warehouse.

### 4. 5. 3 SYSTEM SPECIFICATIONS (Refer to Subsection 3. 2, Functional Specifications)

A. Traffic by Location Daily

In the figures below, the growth factor is included.

| City | Inquiries | Updating | Rush Orders | Minimum Inventory |
|---|---|---|---|---|
| A | 100 | 10 | 110 | 15 |
| B | 90 | 15 | 120 | |
| C | 80 | 20 | 130 | |
| D | 90 | 15 | 120 | |
| E | 40 | 5 | 40 | |
| F | 30 | 7 | 50 | |
| G | 30 | 8 | 60 | |
| | 460 | 80 | 630 | 15 |



SEE FIGURE 103. THE PAPER FLIP BETWEEN THE PERFORATION OF TWO FORMS IS USED FOR INQUIRIES AND STOCK UPDATING. IF THERE IS A RUSH ORDER, THE PRINT LINE FOR THE INPUT IS PLACED AT THE SAME HEIGHT AS THE ARROW. THE INVOICE IS PRINTED AUTOMATICALLY AS AN OUTPUT MESSAGE, AS INDICATED IN FIGURE 103.

4. MINIMUM INVENTORY INDICATION
A. RECEIVER FORMAT
X X X X B – SUPPLIER DATA (25 POS.) – B X X X X
ITEM NUMBER
QUANTITY

Figure 102. Message Formats

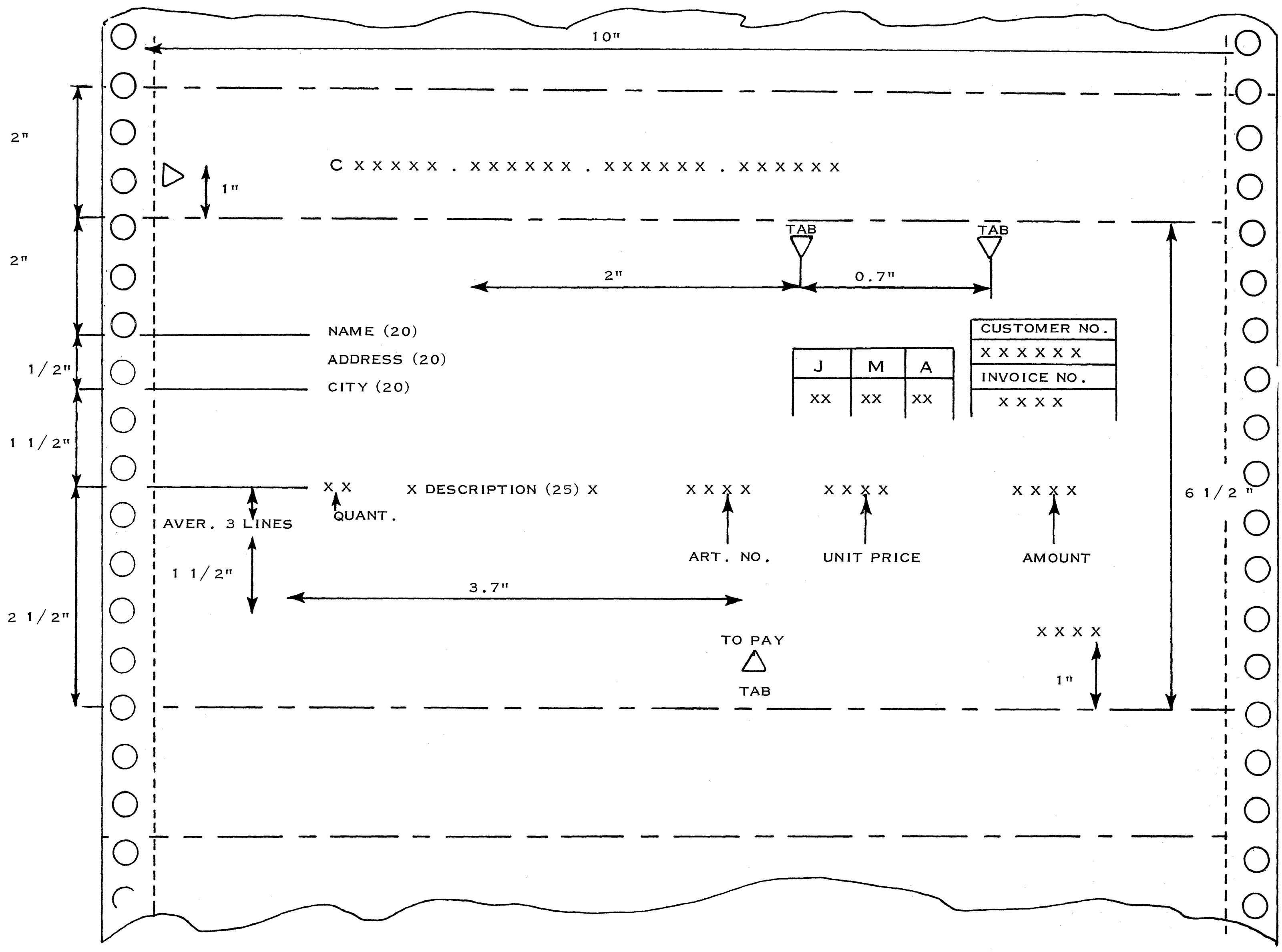
B. Peak hour traffic for the zone warehouses A, B, C, and D is three times that of an average hour; and for zone warehouses E, F, and G, twice that of an average hour.

C. There are no special response requirements.

D. Message formats: see Figure 102 below.

E. Functional Requirements of the Computer Center:

The center's real-time operation consists of the following:



10 CHARACTERS/ INCH
6 LINES/ INCH

Figure 103. Receiver Format

1. Operate all lines simultaneously.

2. Assemble messages from communications lines.

3. Construct and send messages.

4. Detect start and end of messages.

5. Check validity and format.

6. Queue several messages waiting to be processed and allow for buffers.

7. Log input and output messages for stock updating, rush orders, and minimum inventory indications.

8. Code conversion - conversion from terminal code to system code.

9. Failure-recovery in the case of terminal, line, or computer center failure.

10. Handle inquiries, stock updating, and rush orders to the customer and inventory files.

11. Inventory control: update with replenishments, signal minimum inventory.

F. Functional Requirements of the Computer Center, Data Processing or Batch Operation During Tele-processing Hours

- Processing of batch programs which have nothing to do with the Tele-processing application, such as payroll, accounting, statistics, etc.

G. Functional Requirements of the Computer Center Between and After Tele-processing Hours

- Accounts receivable, update daily with new orders and receipts for customer accounts for all seven warehouses;

- Processing and invoicing orders which come in by mail.

H. Logical Files

The logical files are described in the following table.

| Name of File | No. of Records | Size of Records |
|---|---|---|
| Customer | 6,000 | 200 |
| Items | 10,000 | 300 |
| Input Updating | 400 | 20 |
| Output Updating | 400 | 50 |
| Input rush order | 3,150 | 70 |
| Output rush order | 12,600 | 125 |
| Minimum inventory master warehouse | 75 | 50 |
| Minimum inventory zone warehouse | 400 | 50 |

4. 5. 4 PERFORMANCE SPECIFICATIONS

1. Throughput Requirements

The system must be capable of handling the peak-hour traffic requirements

2. Core Storage Requirements

Give core estimates for:

a. Supervisor
b. Programs
c. BTAM
d. Access devices
e. Buffers
f. Total core

3. Give a description of the following:

a. Restart and recovery procedures
b. Backup procedures

4. Reliability Requirements:

Examine whether special requirements are necessary.

5. Operating Requirements:

a. Real time 8 hours per day, 5 days per week.
b. Data processing between and after Tele-processing hours, 3 hours per day, 5 days per week.

### 4.5.5 NECESSARY DISK CAPACITY (Refer to Subsection 3.8, File Organization)

The type of disk file chosen is the 2311.

#### Customer File

For the customer number, we use a 6-digit number. The sixth digit functions as a self-checking number. Therefore, the keys we use for the disk range from 00000 - 99999. However, there are 6,000 customers with whom we do business. The conclusion is that, within the range of keys, there is a high percentage of unused addresses. For this reason, we use the index-sequential technique for the customer file.

For one track entry, we need two records (key length + 10 bytes). In our example, the key is 5 characters. Then, one track can contain 38 records of 15 positions. For the nine prime data tracks (if we reserve one for overflow), we need only 9 track entries (18 records). This means that we need half a track to store the track index. The other half can be used to store data.

The position of each cylinder is:

0.5 track-index track<br>
8.5 prime data tracks<br>
1 overflow track

From the table in Form X20-1705, we find that we can store 12 customer records per track (200 bytes per customer).

To store 6,000 customer records, we need $\frac{6000}{12}$ = 500 tracks. This means (with 8.5 prime data tracks per cylinder) that we need $\frac{500}{8.5}$ = 59 cylinders. To store the cylinder index, we need (key length + 10) bytes per cylinder entry. Therefore, we need 59 records of 15 bytes. That is, $\frac{59}{38}$ = 2 tracks to store the cylinder index.

#### Item File

The item key numbers range from 0000 - 9999. Here it is efficient to use the direct addressing technique. For we have 10,000 records of 300 bytes per item.

From the same table, we find that we can store 9 records of 300 bytes per track. As we have 10,000 records, we need $\frac{10,000}{9}$ = 1,111 tracks = 112 cylinders.

Since we have 200 cylinders available per disk unit, we can store the customer and item file in one unit.

Later, we must decide if this is acceptable with respect to utilization.

#### Input Updating

400 records of 20 bytes per record with direct addressing.

We can store 36 records per track. Thus, we need $\frac{400}{36}$ = 12 tracks = 2 cylinders.

#### Output Updating

400 records of 50 bytes per record with direct addressing.

We can store 27 records per track. Therefore, we need $\frac{400}{27}$ = 15 tracks = 2 cylinders.

#### Input Rush Orders

3,150 records of 70 bytes per record with direct addressing.

We can store 23 records per track. Thus, we need $\frac{3150}{23}$ = 137 tracks = 14 cylinders

#### Output Rush Orders

12,600 records of 125 bytes per record with direct addressing.

We can store 17 records per track. Therefore, we need $\frac{12,600}{17}$ = 742 tracks = 75 cylinders.

#### Minimum Inventory Master Warehouse

75 records of 50 bytes per record with direct addressing.

We can store 28 records per track. Thus, we need $\frac{75}{28}$ = 3 tracks = 1 cylinder.

#### Minimum Inventory Zone Warehouse

400 records of 50 bytes per record with direct addressing.

We can store 28 records per track. Thus, we need $\frac{400}{28}$ = 15 tracks = 2 cylinders.

Total disk capacity needed for logging of the different messages is: 2 + 2 + 14 + 75 + 1 + 2 = 96 cylinders. For this we use a second disk drive.

To decrease seek time, the cylinders of the input and output messages can be located next to each other per message type.

To simplify the design, we have calculated an average seek time of 75 msec. This is a very conservative figure.

### 4. 5. 6 LOAD CALCULATIONS FOR THE CENTRAL COMPUTER (Refer to Subsections 3. 8 and 3. 12)

For the reader's convenience, we repeat the tables we have used for load calculations.

| MESSAGE LENGTH = $\frac{\text{IN + OUT}}{2}$ | MODEL 30 | MODEL 40 | MODEL 50 |
|---|---|---|---|
| 10 CHARACTERS | 11 MS | 6 MS | 4 MS |
| 40 CHARACTERS | 44 MS | 24 MS | 16 MS |
| 100 CHARACTERS | 70 MS | 40 MS | 24 MS |
| 200 CHARACTERS | 140 MS | 80 MS | 48 MS |
| 300 CHARACTERS | 210 MS | 120 MS | 72 MS |
| 400 CHARACTERS | 280 MS | 160 MS | 96 MS |

Figure 104. CPU Time for Message and Line Control Program

| SYSTEM/360 MODEL | AVERAGE INSTRUCTION SPEED IN MICROSECONDS |
|---|---|
| 30 | 35 |
| 40 | 20 |
| 50 | 12 |

Figure 105. Average Instruction Speed

| ACCESS METHOD | MACRO INSTRUCTION | CPU TIME MODEL 30 |
|---|---|---|
| BDAM | READ<br>WRITE | 18.4 MS<br>19.4 MS |
| BISAM | READ<br>WRITE | 37.7 MS<br>87.5 MS |

Figure 106. Macro Execution Time for Accesses

File Response Time (Figures 107 and 108)

The parameter for each graph is based on the number of accesses per second, the number of characters transferred, and the number of devices on the channel.

Moreover, we must consider that the total average number of accesses for this example is less than two per second. This can be checked when we know the traffic during the peak period and the number of accesses per type of transaction. For a file response time of less than 150 msec and one access device, the curves are vague. Therefore, we use 150 msec as a minimum figure. The total processing time for one access is the CPU time (read or write) plus the file response time.

Inquiries

Figure 109 shows the different steps in processing this type of message.

Translate and Analyze the Message (Figure 109, Blocks 1 and 2, and Figure 110)

These blocks represent the BTAM program which reads, checks and analyzes the message before actual processing. The length of this type of message is five characters, represented as follows:

Ixxxx

This takes about 3 milliseconds, plus 7 ms for a BTAM read. The total in + out message and line control takes 17 ms.

In this case, there is no need to write the input message on disk, for the following reasons:

1. For the inquiry type, we do not need error recovery procedures because nothing is changed in the files.

2. With this type of message, we always work in conversational mode, so that there is no need to empty the buffer before the message is processed and the answer can be sent back.

Read Item Record (Figures 111 and 109, Block 3)

For the item file we used the direct access method. We must examine the timing considerations shown in Figure 111.

Processing (Figures 112 and 109, Block 4)

Let us suppose that about 500 instructions are necessary for the actual processing. We have observed

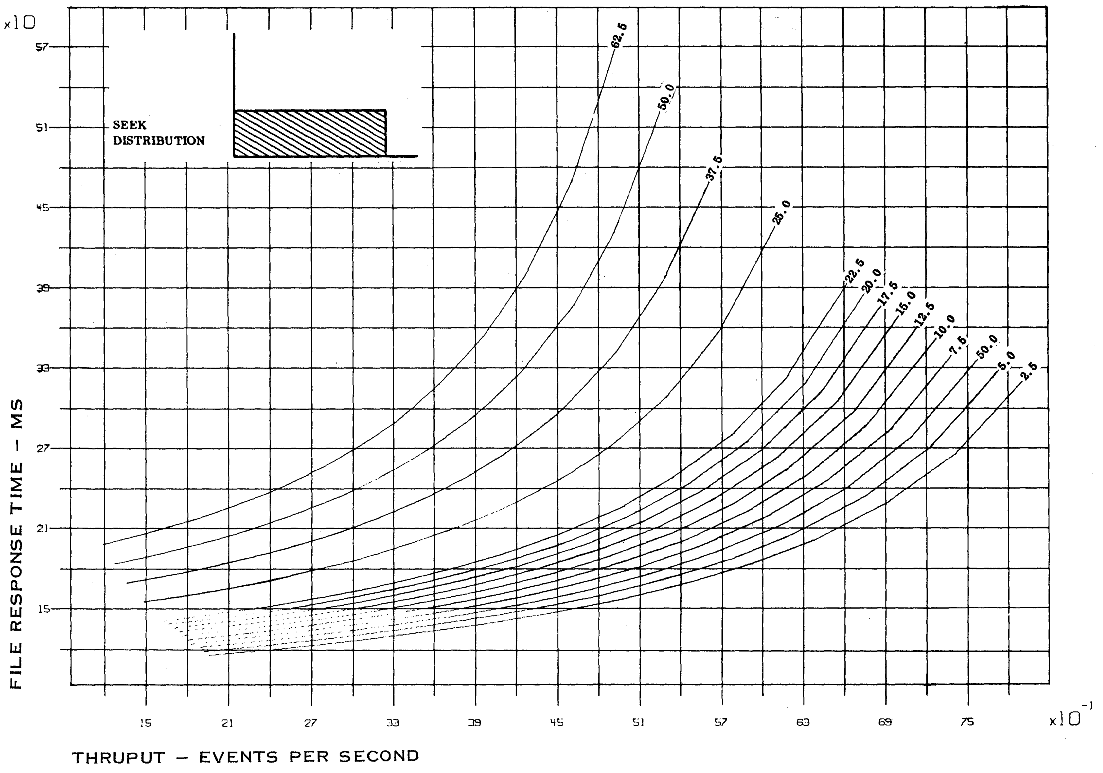


Figure 107. File Response Time: System with One Access Device

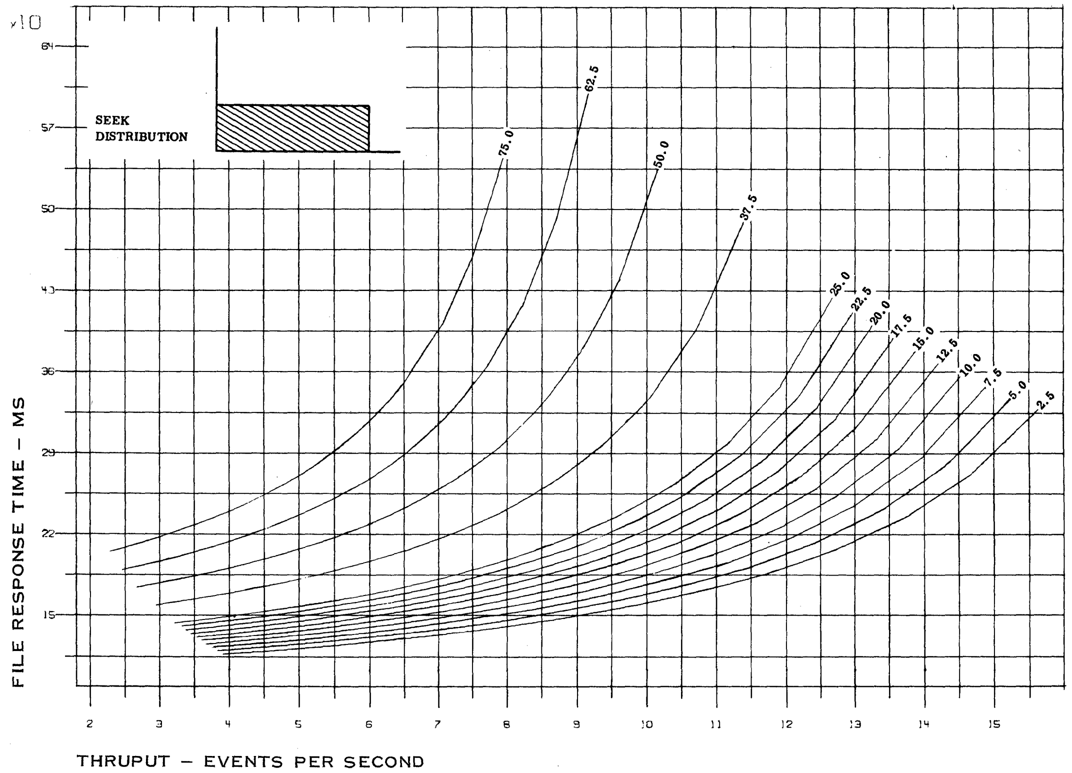
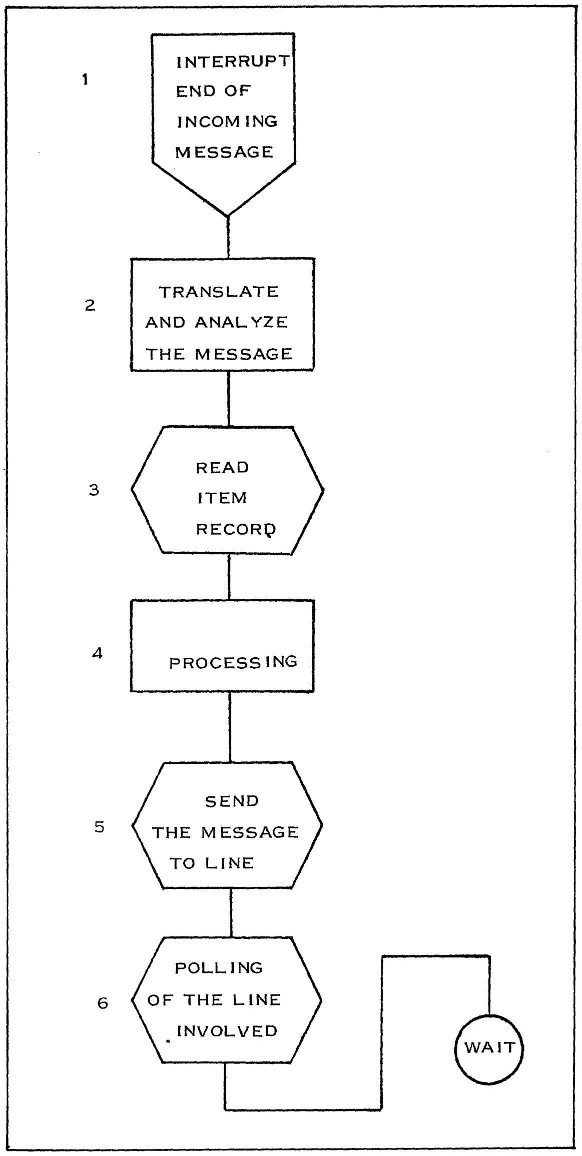


Figure 108. File Response Time: System with Two Access Devices

Note: Each curve represents different values for total channel service time.



1
INTERRUPT END OF INCOMING MESSAGE
2
TRANSLATE AND ANALYZE THE MESSAGE
3
READ ITEM RECORD
4
PROCESSING
5
SEND THE MESSAGE TO LINE
6
POLLING OF THE LINE INVOLVED
WAIT

Figure 109. Flowchart: Inquiry Processing

| INPUT MESSAGE CONTROL | 2311 | MPX CHANNEL | MESSAGE LINE CTL. CPU TIME | PROCESSING |
|---|---|---|---|---|
| | | NEGLIGIBLE | 10 MSEC | 10 MSEC |

Figure 110. Input Message: Timing

| | 2311-1 | SEL. CHANNEL | CPU | PROC. |
|---|---|---|---|---|
| | (MS) | (MS) | (MS) | (MS) |
| PROCESSING TIME – (READ BDAM) | 18.4 | 18.4 | 18.4 | 18.4 |
| SEEK TIME | 75 | | | 150 |
| ROTATIONAL DELAY | 12.5 | 12.5 | | |
| RECORD TIMES 300 X 7 X $10^{-3}$ | 2.1 | 2.1 | | |
| TOTAL | 108 | 33 | 18.4 | 168.4 * |

* Includes time waiting in a disk queue. Refer to Figures 107 and 108.

Figure 111. Read Item Record: Timing

| | 2311 | SEL. CHANNEL | CPU | PROC. |
|---|---|---|---|---|
| PROCESSING 500 X 35 μs | | | 18 MSEC | 18 MSEC |

Figure 112. Processing: Timing

that the time per instruction is about 35 μs for a Model 30.

Send Message to the Line (Figures 113 and 109, Blocks 5 and 6)

This is executed by the message control task (BTAM). The length of the output message is 25 characters. This means about 14 msec for message and line control, plus 7 ms for a BTAM write.

| OUTPUT MESSAGE CONTROL | 2311 | MPX CHANNEL | CPU | PROCESSING |
|---|---|---|---|---|
| | | NEGLIGIBLE | 21 MSEC. | 21 MSEC. |

Figure 113. Message Control: Timing

## Stock Updating

Figure 114 shows the required steps for processing this type of message.

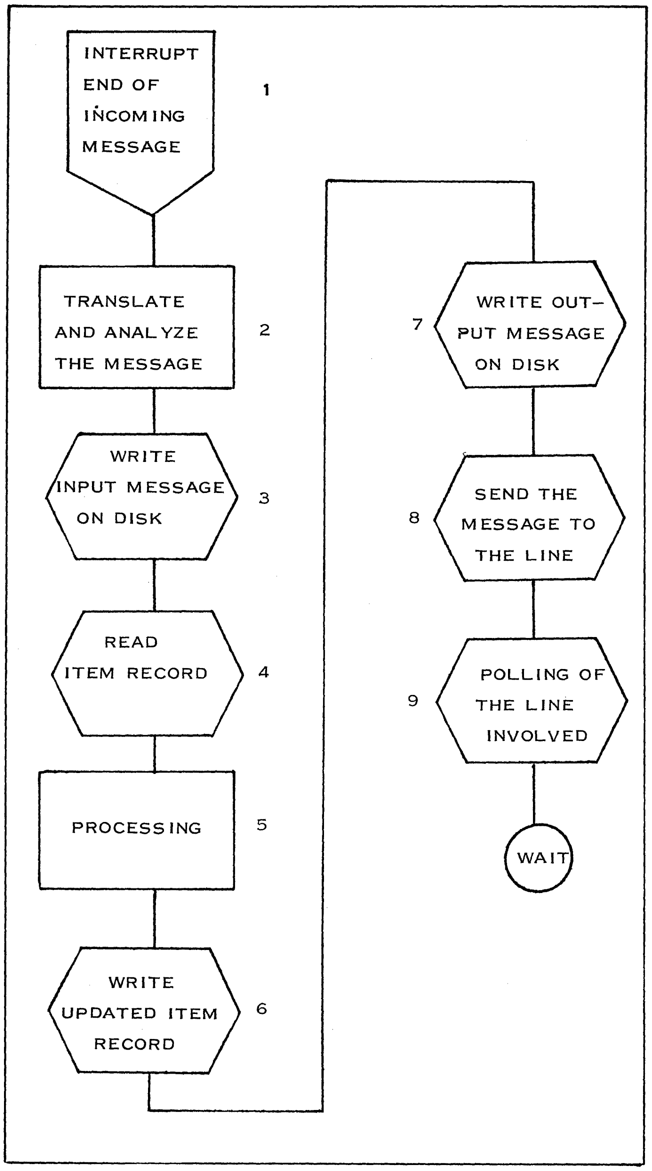

INTERRUPT END OF INCOMING MESSAGE
1
TRANSLATE AND ANALYZE THE MESSAGE
2
WRITE INPUT MESSAGE ON DISK
3
READ ITEM RECORD
4
PROCESSING
5
WRITE UPDATED ITEM RECORD
6
7
WRITE OUT-PUT MESSAGE ON DISK
8
SEND THE MESSAGE TO THE LINE
9
POLLING OF THE LINE INVOLVED
WAIT

Figure 114. Flowchart: Stock Updating

Translate and Analyze the Message (Figures 115 and 114, Blocks 1 and 2)

The length of this input message is 9 characters:

S x x x x x x x x

This takes about 5 milliseconds, plus 7 ms for a BTAM read. The total in + out message and line control time takes 22 ms.

| | 2311 | CHANNEL | CPU | PROC. |
|---|---|---|---|---|
| INPUT MESSAGE CONTROL | | | 12 MSEC | 12 MSEC |

Figure 115. Input Analysis and Translation of Message: Timing

Write Input Message on Disk (Figures 116 and 114, Block 3)

For the following reasons, input messages are written on disk:

1. For processing this data afterwards in a batch program,

2. For recovery procedure.

In case of CPU failure, we would like to be able to reconstruct the situation at the moment of the breakdown. This can be done by logging the input and output messages on disk.

For this, we use direct addressing, with fixed record length, while we keep in core a table of the input messages to be processed and their corresponding disk addresses. For the addition of the necessary extra header information, we choose a record length of 20 bytes per input message.

| | 2311-1 | CHANNEL | CPU | PROCESSING |
|---|---|---|---|---|
| | MS | MS | MS | MS |
| PROCESSING TIME – WRITE (BDAM) | 19.4 | 19.4 | 19.4 | 19.4 |
| SEEK TIME | 75 | | | |
| ROTATIONAL DELAY | 12.5 | 12.5 | | 150 |
| DATA TRANSFER $7 \cdot 20 \times 10^{-3}$ | 0.2 | 0.2 | | |
| WRITE CHECK | 25 | 25 | | |
| TOTAL | 132.1 | 57.1 | 19.4 | 169.4* |

*Includes time waiting in a disk queue. Refer to Figures 107 and 108.

Figure 116. Writing Input Message on Disk: Timing

Read Item Record (Figures 117 and 114, Block 4)

We use the direct access method for the item file. The record length is 300 bytes per item. We are concerned with the timings shown in Figure 117.

| | 2311-II | CHANNEL | CPU | PROC. |
|---|---|---|---|---|
| | (MS) | (MS) | (MS) | (MS) |
| PROCESSING TIME – READ (BDAM) | 18.4 | 18.4 | 18.4 | 18.4 |
| SEEK TIME | 75 | | | |
| ROTATIONAL DELAY | 12.5 | 12.5 | | 150 |
| DATA TRANSFER 7 X 300 X $10^{-3}$ | 2.1 | 2.1 | | |
| TOTAL | 108 | 33 | 18.4 | 168.4* |

* Includes time waiting in a disk queue. Refer to Figures 107 and 108.

Figure 117. Read Item Record: Timing

Processing (Figures 118 and 114, Block 5)

Let us suppose that about 750 instructions are necessary for actual message processing. We have observed that the execution time per instruction is about 35 µs for the Model 30.

| | 2311 | CHANNEL | CPU | PROC. |
|---|---|---|---|---|
| PROCESSING<br>750 X 35 µS | | | 26 MSEC | 26 MSEC |

Figure 118. Processing: Timing

Write Updated Item Record (Figures 119 and 114, Block 6)

The time we need for this is shown in Figure 119.

| | 2311-II | CHANNEL | CPU | PROC. |
|---|---|---|---|---|
| | (MS) | (MS) | (MS) | (MS) |
| PROCESSING TIME – WRITE (BDAM) | 19.4 | 19.4 | 19.4 | 19.4 |
| SEEK TIME | 75 | | | |
| ROTATIONAL DELAY | 12.5 | 12.5 | | |
| DATA TRANSFER 7 X 300 X $10^{-3}$ | 2.1 | 2.1 | | 150 |
| WRITE CHECK | 25 | 25 | | |
| TOTAL | 134 | 59 | 19.4 | 169.4* |

*Includes time waiting in a disk queue. Refer to Figures 107 and 108.

Figure 119. Write Updated Item Record: Timing

Write Output Message on Disk (Figures 120 and 114, Block 7)

For the error recovery procedure, we write this type of output message on disk. The length of this message is 35 positions.

For the additional header information, we choose records of 50 bytes per message. For logging this, we also use the direct addressing technique.

| | 2311-I | CHANNEL | CPU | PROC. |
|---|---|---|---|---|
| | (MS) | (MS) | (MS) | (MS) |
| PROCESSING TIME – WRITE (BDAM) | 19.4 | 19.4 | 19.4 | 19.4 |
| SEEK TIME | 75 | | | |
| ROTATIONAL DELAY | 12.5 | 12.5 | | 170 |
| DATA TRANSFER 7 X 50 X $10^{-3}$ | 0.4 | 0.4 | | |
| WRITE CHECK | 25 | 25 | | |
| TOTAL | 132.3 | 57.3 | 19.4 | 169.4* |

* Includes time waiting in a disk queue. Refer to Figures 107 and 108.

Figure 120. Write Output Message on Disk: Timing

Send the Message to the Line (Figures 121 and 114, Blocks 8 and 9)

This will be executed by the message control task (BTAM). The length of the output message is 35 positions. This means about 17 msec for message and line control, plus 7 ms for a BTAM write.

| | 2311-II | CHANNEL | CPU | PROC. |
|---|---|---|---|---|
| OUTPUT MESSAGE CONTROL | | | 24 MSEC | 24 MSEC |

Figure 121. Send the Message to the Line: Timing

Rush Order Entry (Figure 122)

Figure 122 shows the steps required for processing this type of message.

Translate and Analyze the Message (Figures 123 and 122, Blocks 1 and 2)

The average length of the input message is 28 characters:

C x x x x x . x x x x x x . x x x x x x .

This takes about 10 msec, plus 7 ms for a BTAM read. The total in + out message and line control time takes 81 ms.

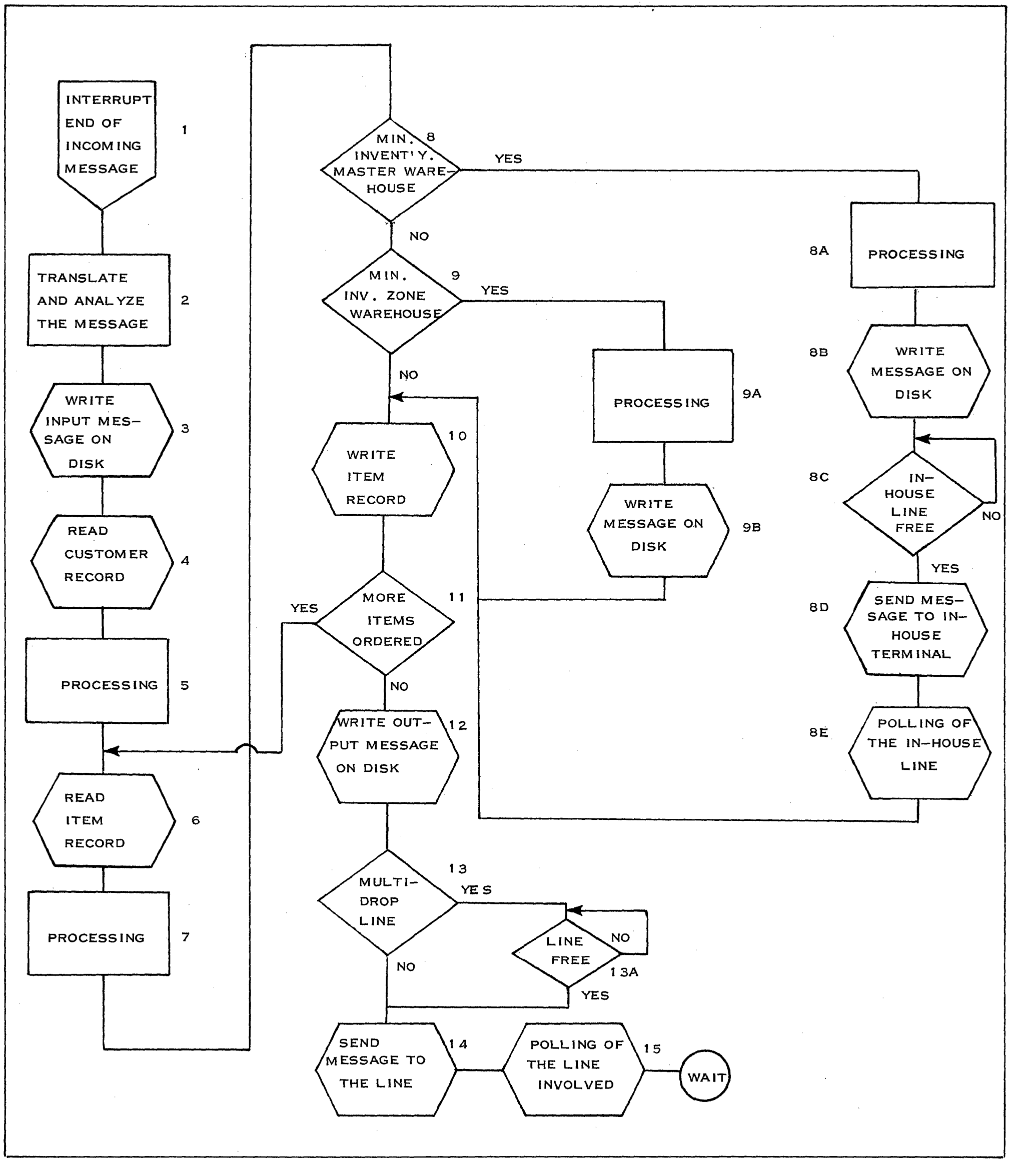


Figure 122. Flowchart: Rush Order Entry

| | 2311 | CHANNEL | CPU | PROC. |
|---|---|---|---|---|
| INPUT MESSAGE CONTROL | | | 17 MSEC | 17 MSEC |

Figure 123. Input Translation and Analysis of Message: Timing

Write Input Message on Disk (Figures 124 and 122, Block 3)

Input messages are written on disk, for the following reasons:

1. Processing this data later in a batch program (i. e. updating customer records);

2. Recovery procedures.

For this logging, we use the direct addressing techniques with fixed, unblocked records. As the maximum message length can be 50 characters, we choose for the additional header information records of 70 bytes.

| | 2311-I | CHANNEL | CPU | PROC. |
|---|---|---|---|---|
| | (MS) | (MS) | (MS) | (MS) |
| PROCESSING TIME – WRITE (BDAM) | 19.4 | 19.4 | 19.4 | 19.4 |
| SEEK TIME | 75 | | | 150 |
| ROTATIONAL DELAY | 12.5 | 12.5 | | |
| DATA TRANSFER $7 . 70 . 10^{-3}$ | 0.5 | 0.5 | | |
| WRITE CHECK | 25 | 25 | | |
| TOTAL | 132.4 | 57.4 | 19.4 | 169.4* |

* Includes disk queue waiting time.

Figure 124. Writing Input Message on Disk: Timing

Read Customer Record (Figures 125 and 122, Block 4)

This record is read for printing the name, address, and city of the customer on the invoice.

The customer file is organized via the index-sequential technique. In this example, we need two accesses per record. However, if the cylinder index is in core, we need only one access.

The record length is 200 bytes per customer.

| | 2311-II | CHANNEL | CPU | PROC. |
|---|---|---|---|---|
| | (MS) | (MS) | (MS) | (MS) |
| 1ST ACCESS TO CYLINDER INDEX | | | | |
| PROCESSING TIME – READ (BDAM) | 18.4 | 18.4 | 18.4 | 18.4 |
| SEEK TIME | 75 | | | 150 |
| ROTATIONAL DELAY | 12.5 | 12.5 | | |
| DATA TRANSFER $20 . 7 . 10^{-3}$ | 0.2 | 0.2 | | |
| 2ND ACCESS TO TRACK INDEX AND RECORD | | | | |
| PROCESSING TIME – READ (BISAM) | 37.7 | 37.7 | 37.7 | 37.7 |
| SEEK TIME | 75 | | | 170 |
| ROTATIONAL DELAY | 12.5 | 12.5 | | |
| 1 REVOLUTION | 25 | 25 | | |
| DATA TRANSFER $7 . 200 . 10^{-3}$ | 1.4 | 1.4 | | |
| TOTAL | 257.1 | 107.7 | 56.1 | 356.1* |

* Includes disk queue waiting time.

Figure 125. Reading Customer Record: Timing

Processing (Figures 126 and 122, Block 5)

Let us suppose that for all the processing and decision blocks indicated in the flowchart 1,500· instructions are necessary. The execution time per instruction is about 35 µs for the Model 30.

| | 2311 – II | CHANNEL | CPU | PROC. |
|---|---|---|---|---|
| PROCESSING 1,500 X 35 µS | | | 53 MS | 53 MS |

Figure 126. Processing: Timing

Read Item Record (Figures 127 and 122, Block 6)

We use the direct access method for the item file. The record length is 300 bytes per item. We are concerned with the timings shown in Figure 127.

Processing (Figures 122, Blocks 7 to 9B)

Minimum inventory master warehouse }
Minimum inventory zone warehouse }

The blocks mentioned above have already been included in the 1,500 processing instructions.

| | 2311-II | CHANNEL | CPU | PROC. |
|---|---|---|---|---|
| | (MS) | (MS) | (MS) | (MS) |
| PROCESSING TIME – READ (BDAM) | 18.4 | 18.4 | 18.4 | 18.4 |
| SEEK TIME | 75 | | | 150 |
| ROTATIONAL DELAY | 12.5 | 12.5 | | |
| DATA TRANSFER 7 X 300 X $10^{-3}$ | 2.1 | 2.1 | | |
| TOTAL | 108 | 33 | 18.4 | 168.4* |

* Includes disk queue waiting time.

Figure 127. Read Item Record: Timing

Write Item Record (Figures 128 and 122, Block 10)

The time required for this is shown in Figure 128.

| | 2311-II | CHANNEL | CPU | PROC. |
|---|---|---|---|---|
| | (MS) | (MS) | (MS) | (MS) |
| PROCESSING TIME – WRITE (BDAM) | 19.4 | 19.4 | 19.4 | 19.4 |
| SEEK TIME | 75 | | | 170 |
| ROTATIONAL DELAY | 12.5 | 12.5 | | |
| DATA TRANSFER 7 X 300 X $10^{-3}$ | 2.1 | 2.1 | | |
| WRITE CHECK | 25 | 25 | | |
| TOTAL | 134 | 60 | 19.4 | 169.4* |

* Includes disk queue waiting time.

Figure 128. Write Item Record: Timing

More Items Ordered (Figure 122, Block 11)

These are included in the 1,500 instructions previously mentioned.

Write Output Message on Disk (Figures 129 and 122, Block 12)

This output message is written on disk, for the following reasons:

1. Error recovery procedures;
2. The message has to be segmented;
3. Queuing of messages for the multi-drop line.

We use the direct addressing technique. One line of the invoice cannot exceed 55 characters (including idle characters). We store two lines per record. Then we need records of 125 bytes for the additional header information.

| | 2311-I | CHANNEL | CPU | PROC. |
|---|---|---|---|---|
| | (MS) | (MS) | (MS) | (MS) |
| PROCESSING TIME – WRITE (BDAM) | 19.4 | 19.4 | 19.4 | 19.4 |
| SEEK TIME | 75 | | | 150 |
| ROTATIONAL DELAY | 12.5 | 12.5 | | |
| DATA TRANSFER 7 X 125 X $10^{-3}$ | 0.9 | 0.9 | | |
| WRITE CHECK | 25 | 25 | | |
| TOTAL | 132.8 | 57.8 | 19.4 | 169.4* |

*Includes disk queue waiting time.

Figure 129. Writing Output Messages on Disk: Timing

Multi-drop Line (Figure 122, Blocks 13 and 13A)

This is included in the 1,500 instructions previously mentioned.

Send the Message to the Line (Figures 130 and 122, Blocks 14 and 15)

This is executed by the message control task (BTAM). The average length of the output message is about 250 characters. This means about 71 msec for message and line control, plus 7 ms for a BTAM write.

|  | 2311 | CHANNEL | CPU | PROC. |
|---|---|---|---|---|
| OUTPUT MESSAGE CONTROL |  |  | 78 MS | 78 MS |

Figure 130. Sending Message to the Line: Timing

Minimum Inventory Master Warehouse Exceeded (Figure 122, Blocks 8 to 8E)

In this case, the order program will be lengthened with blocks 8A to 8E. The total average daily traffic is 630 rush orders, against 15 minimum inventory indication messages. As there is an average of 3 items per day, we have 1,890 item lines against 15 minimum inventory messages. The effect on the average increase in processing time is $\frac{15}{1890}$ x 100% = 8% of the timing.

Processing (Figure 122, Block 8A)

This is included in the 1,500 instructions.

Write Message on Disk (Figures 131 and 122, Block 8B)

This is done for the following reasons:

1. Error recovery
2. Processing this data later in a batch program.

|  | 2311-I | CHANNEL | CPU | PROC. |
|---|---|---|---|---|
|  | (MS) | (MS) | (MS) | (MS) |
| PROCESSING TIME – WRITE (BDAM) | 19.4 | 19.4 | 19.4 | 19.4 |
| SEEK TIME | 75 |  |  |  |
| ROTATIONAL DELAY | 12.5 | 12.5 |  | 150 |
| DATA TRANSFER 7 X 50 X $10^{-3}$ | 0.4 | 0.4 |  |  |
| WRITE CHECK | 25 | 25 |  |  |
| TOTAL | 132.3 | 57.3 | 19.4 | 169.4 |
| 0.8% | 1.1 | 0.5 | 0.2 | 1.4 |

Figure 131. Writing Message on Disk: Timing

The message length is 35 positions. We choose records of 50 bytes per message for the additional header information.

Send Message to In-House Terminal (Figures 132 and 122, Blocks 8D and 8E)

This is executed by the message control program (BTAM). The length of the output message is 35 positions. This takes about 39 msec.

|  | 2311 | CHANNEL | CPU | PROC. |
|---|---|---|---|---|
| OUTPUT MESSAGE CONTROL |  |  | 39 MS | 39 MS |
| 0.8 % |  |  | 0.3 MS | 0.3 MS |

Figure 132. Sending Message to the In-House Terminal: Timing

Minimum Inventory Zone Warehouse Exceeded (Figure 122, Blocks 9)

The total average daily traffic is 630 rush orders against 80 updating messages. As there is an average of 3 items per order, we have 1,890 item lines against 80 updating messages, so that the average processing time of the order program is increased $\frac{80}{1890}$ x 100% = 4.3% of the timing calculated for a normal order. In this case, the order program is lengthened with blocks 9A and 9B of Figure 122.

|  | 2311 | CHANNEL | CPU | PROC. |
|---|---|---|---|---|
| PROCESSING TIME – WRITE (BDAM) | 18.4 | 18.4 | 18.4 | 18.4 |
| SEEK TIME | 74 |  |  |  |
| ROTATIONAL DELAY | 12.5 | 12.5 |  |  |
| DATA TRANSFER 7 X 50 X $10^{-3}$ | 0.4 | 0.4 |  |  |
| WRITE CHECK | 25 | 25 |  |  |
| TOTAL | 132.3 | 57.3 | 18.4 | 168.4 |
| 4.3% | 5.7 | 2.5 | 0.7 | 7.3 |

Figure 133. Writing Message on Disk: Timing

## Processing (Figure 122, Block 9A)

This is included in the 1,500 instructions previously mentioned.

## Write Message on Disk

See Figures 133 and 122, Block 9B.

| MAIN STEPS PER TYPE OF TRANSACTION | 2311-1 IN MS | 2311-2 IN MS | CHANNEL IN MS | CPU IN MS | TOTAL PROCESSING TIME (INT. RESP. TIME) IN MS |
|---|---|---|---|---|---|
| INQUIRY | | | | | |
| INPUT MESSAGE CONTROL | | | | 10 | 10 |
| READ ITEM RECORD | | 108 | 32 | 18.4 | 168.4 |
| PROCESSING | | | | 18 | 18 |
| SEND MESSAGE TO THE LINE | | | | 21 | 21 |
| | | 108 | 32 | 67.4 | 217.4 |
| STOCK UPDATING | | | | | |
| INPUT MESSAGE CONTROL | | | | 12 | 12 |
| WRITE INPUT MESSAGE ON DISK | 132.1 | | 57.1 | 19.4 | 169.4 |
| READ ITEM RECORD | | 108 | 33 | 18.4 | 168.4 |
| PROCESSING | | | | 26 | 26 |
| WRITE ITEM RECORD | | 134 | 59 | 19.4 | 169.4 |
| WRITE OUTPUT MESSAGE ON DISK | 132.3 | | 57.3 | 19.4 | 169.4 |
| SEND MESSAGE TO THE LINE | | | | 24 | 24 |
| | 264.4 | 242 | 206.4 | 138.6 | 738.6 |
| RUSH ORDERS | | | | | |
| INPUT MESSAGE CONTROL | | | | 17 | 17 |
| WRITE INPUT MESSAGE ON DISK | 132.4 | | 57.4 | 19.4 | 169.4 |
| READ CUSTOMER RECORD | | 257.1 | 107.7 | 56.1 | 356.1 |
| PROCESSING | | | | 53 | 53 |
| READ ITEM RECORD | | 108 | 33 | 18.4 | 168.4 |
| WRITE ITEM RECORD | | 134 | 60 | 19.4 | 169.4 |
| READ ITEM RECORD | | 108 | 33 | 18.4 | 168.4 |
| WRITE ITEM RECORD | | 134 | 60 | 19.4 | 169.4 |
| READ ITEM RECORD | | 108 | 33 | 18.4 | 168.4 |
| WRITE ITEM RECORD | | 134 | 60 | 19.4 | 169.4 |
| WRITE OUTPUT MESSAGE ON DISK | 132.8 | | 57.8 | 19.4 | 169.4 |
| SEND MESSAGE TO THE LINE | | | | 78 | 78 |
| WRITE MESSAGE ON DISK | 1.1 | | 0.5 | 0.2 | 1.4 |
| SEND MESSAGE TO IN-HOUSE TERMINAL | | | | 0.3 | 0.3 |
| WRITE MESSAGE ON DISK | 5.7 | | 2.7 | 0.8 | 7.3 |
| | 272 | 983.1 | 505.1 | 357.6 | 1,865.3 |

Figure 134. Summary of Calculations

## 4. 5. 7 CALCULATION OF THE DEVICE LOAD

Calculation of the device load is shown in Figure 135.

| DEVICE | TYPE OF TRANSACTION | NO. OF TRANSACTIONS PER PEAK PERIOD<br>A | DEVICE TIME PER TRANSACTION<br>B | ABSOLUTE LOAD<br>A X B | PEAK PERIOD<br>C | RELATIVE LOAD<br>$\frac{\sum A \text{ X } B}{C}$ |
|---|---|---|---|---|---|---|
| 2311-1 | INQUIRY | 167 | | | | |
| | STOCK UPDATING | 33 | 264.4 MS | 8.7 SEC. | | $\frac{69.3}{3,600} =$ |
| | RUSH ORDERS | 223 | 272.0 | 60.6 | | |
| | | | | 69.3 SEC. | 3,600 SEC. | 1.9 % |
| 2311-2 | INQUIRY | 167 | 108.0 MS | 18.0 SEC. | | |
| | STOCK UPDATING | 33 | 242.0 | 8.0 | | $\frac{245.0}{3,600} =$ |
| | RUSH ORDERS | 223 | 983.1 | 219.0 SEC. | | |
| | | | | 245.0 SEC. | 3,600 SEC. | 6.8 % |
| CHANNEL | INQUIRY | 167 | 32.0 MS | 5.4 SEC. | | |
| | STOCK UPDATING | 33 | 206.4 | 6.8 | | $\frac{125.1}{3,600} =$ |
| | RUSH ORDERS | 223 | 505.1 | 112.9 | | |
| | | | | 125.1 SEC. | 3,600 SEC. | 3.5 % |
| CPU | INQUIRY | 167 | 49.4 MS | 8.3 SEC. | | |
| | STOCK UPDATING | 33 | 138.6 | 4.6 | | $\frac{92.7}{3,600} =$ |
| | RUSH ORDERS | 223 | 357.6 | 79.8 | | |
| | | | | 92.7 SEC. | | 2.6 % |

Figure 135. Calculation of the Device Load

Besides the CPU utilization already calculated, there is also a CPU load caused by the MPX channel interference, due to data transfer, interrupts, polling, and negative polling. Use the estimates given in Figure 136 for CPU interference due to negative polling, since this is the prime contributor to MPX channel interference.

| SYSTEM/360<br>MODEL | CPU TIME<br>PER "POLLING" |
|---|---|
| 30 | 3.5 MS |
| 40 | 3.5 MS |
| 50 | 1.8 MS |

Figure 136. CPU Time Necessary for Polling

We can calculate that the average load of the five lines in our example is about 65%. This means that 35% of the time we have negative polling.

If we have two pollings per second, per line, we get $2 \times 5.5 \times 5 \times 0.35 = \frac{\text{20 milliseconds.}}{\text{second}}$

This means 2% CPU utilization.

The conclusion is that the additional load to the CPU of the different devices due to the CBS application is very small.

## 4. 5. 8 CALCULATION OF TERMINAL TIME PER TRANSACTION (Figure 137)

Let us suppose that the keying speed of the operator is 2 characters/second. This means $\frac{1000}{2} = 500$ msec/character.

For each input message we add, for safety, two additional seconds for operator time per transaction. This time is wasted by paper shuffling, operator response to polling, etc.

From the calculations made in Terminal Load Calculations, subsection 4. 2. 4, we derived the following results for printing:

- Reasonable form layout, about 10 characters/second, or 100 msec/character

- Inefficient form layout, about 8 characters/second, or 125 msec/character.

With the aid of these figures, we can calculate the timings for the different transaction types.

With the intermediate paper tape reader and punch units, we can key in the data in a home-loop operation and send it to the computer in a line-loop operation. Also, the invoice can be printed out in a home-loop operation. In that case, we do not have to insert the idle characters. This method decreases the line load. Conversational mode is used at the multi-drop terminals for inquiry and stock updating messages only. Rush orders are sent to the computer by the paper tape reader, and the output messages (invoices) are punched in paper tape.

The polling sequence will be:

K 1, K 2, K 3, PR 1, PR 2, PR 3

(K = keyboard; PR = paper tape reader)

To give the highest priority to the keyboard, the following arrangements are made. As soon as one of the paper tape readers has sent a message to the computer, the polling sequence starts again with K 1, K 2, K 3. When one invoice has been sent to a multi-drop terminal, there is, again, polling of the keyboard.

## 4. 5. 9 CALCULATION OF THE PEAK LOAD (Figure 138)

From the daily traffic described in the system specifications, we can derive the figures shown in Figure 138.

## 4. 5. 10 CALCULATION OF THE TERMINAL LOAD DURING PEAK PERIOD

For the calculation of the terminal load during the peak period, see Figure 139.

The line load for the point-to-point connections equals the terminal load.

The recommended maximum terminal load is 75%.

This ranges, in our calculations, from 61% to 75%.

The most heavily loaded terminal is in zone warehouse C, at 72%. This means that, with this load, the mean waiting time at the terminal caused by queuing is $\frac{0.72}{1 - 0.72} = 2.6$ times the average time per transaction. The average time per transaction in zone warehouse C is:

$$\frac{30 \times 7.8 + 9 \times 11.3 + 51 \times 44.0}{90} =$$

$$\frac{234 + 101.7 + 2244.0}{90} =$$

$$\frac{2579.7}{90} = 28.7 \text{ seconds.}$$

| TYPE OF TRANSACTION | NO. OF CHAR. | MEDIA | | EFFECTIVE SPEED IN CPS | TIME CHAR. | OPERATOR TIME | TOTAL TIME | INTERNAL RESPONSE TIME | TERMINAL TIME PER TRANSACTION |
|---|---|---|---|---|---|---|---|---|---|
| | | INPUT | OUTPUT | | | | | | |
| | | | | | MS | MS | MS | MS | MS |
| INQUIRY | 6 | KEYBOARD | | 2 | 500 | 2,000 | 5,000 | | |
| | 25 | | PRINTER | 10 | 100 | | 2,500 | 217.4 | 7,717 |
| STOCK UPDATING | 10 | KEYBOARD | | 2 | 500 | 2,000 | 7,000 | | |
| | 35 | | PRINTER | 10 | 100 | | 3,500 | 738.6 | 11,239 |
| RUSH ORDERS | 29 | KEYBOARD | | 2 | 500 | 2,000 | 16,500 | | |
| | 206 | | PRINTER | 8 | 125 | | 25,750 | 1,865 | 44,115 |
| | 29 | PAPER TAPE | | 10 | 100 | | 2,900 | | |
| | 206 | | PAPER TAPE | 10 | 100 | | 20,600 | 1,865 | 25,365 |
| MIN. INVENTORY | 33 | | PRINTER | 10 | 100 | | 3,500 | | 3,500 |

Figure 137. Calculation of Terminal Time Per Transaction

| | NUMBER OF MESSAGES PER HOUR | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | AVERAGE TRAFFIC | | | | PEAK TRAFFIC | | | |
| CITY | INQUIRIES | UPDATING | RUSH ORDERS | MIN. INVENTORY | INQUIRIES | UPDATING | RUSH ORDERS | MIN. INVENTORY |
| A | 13 | 2 | 14 | 2 | 39 | 6 | 42 | 6 |
| B | 12 | 2 | 15 | | 36 | 6 | 45 | |
| C | 10 | 3 | 17 | | 30 | 9 | 51 | |
| D | 12 | 2 | 15 | | 36 | 6 | 45 | |
| E | 5 | 1 | 5 | | 10 | 2 | 10 | |
| F | 4 | 1 | 7 | | 8 | 2 | 14 | |
| G | 4 | 1 | 8 | | 8 | 2 | 16 | |
| | 60 | 12 | 81 | 2 | 167 | 33 | 223 | 6 |

Figure 138. Calculation of the Peak Load

| TYPE OF CITY TRANSACTIONS | NO. OF TRANS. PER PEAK PERIOD<br>A | TIME PER TRANSACTION IN SECONDS *<br>B | A X B | PEAK PERIOD IN SECONDS | TERMINAL LOAD<br>$\Sigma \frac{A\ X\ B}{C}$ |
|---|---|---|---|---|---|
| A. INQUIRY | 39 | 7.8 | 304.2 | | |
| STOCK UPDATING | 6 | 11.3 | 67.8 | | |
| RUSH ORDERS | 42 | 44.2 | 1,865.0 | | 2,249.0 = |
| MIN. INVENTORY | 6 | 3.5 | 21.0 | | 3,600 |
| | | | 2,258.0 | 3,600 | 61 % |
| B. INQUIRY | 36 | 7.8 | 280.8 | | |
| STOCK UPDATING | 6 | 11.3 | 67.8 | | 2,337.6= |
| RUSH ORDERS | 45 | 44.2 | 1,989.0 | | 3,600 |
| | | | 2,337.6 | 3,600 | 65 % |
| C. INQUIRY | 30 | 7.8 | 234.0 | | |
| STOCK UPDATING | 9 | 11.3 | 101.7 | | 2,589.9= |
| RUSH ORDERS | 51 | 44.2 | 2,254.2 | | 3,600 |
| | | | 2,589.9 | 3,600 | 72 % |
| D. INQUIRY | 36 | 7.8 | 280.8 | | |
| STOCK UPDATING | 6 | 11.3 | 67.8 | | 2,328.6= |
| RUSH ORDERS | 45 | 44.2 | 1,980.0 | | 3,600 |
| | | | 2,328.6 | 3,600 | 65 % |
| E. INQUIRY | 10 | 7.8 | 78.0 | | 352.6= |
| STOCK UPDATING | 2 | 11.3 | 22.6 | | 3,600 |
| RUSH ORDERS | 10 | 25.4 | 252.0 | | 10 % |
| | | | 352.6 | 3,600 | |
| F. INQUIRY | 8 | 7.8 | 62.4 | | |
| STOCK UPDATING | 2 | 11.3 | 22.6 | | 440.6= |
| RUSH ORDERS | 14 | 25.4 | 355.6 | | 3,600 |
| | | | 440.6 | 3,600 | 13 % |
| G. INQUIRY | 8 | 7.8 | 62.4 | | |
| STOCK UPDATING | 2 | 11.3 | 22.6 | | 491.2= |
| RUSH ORDERS | 16 | 25.4 | 406.4 | | 3,600 |
| | | | 491.4 | 3,600 | 14 % |

*Rounded upward

Figure 139. Calculation of Terminal Load During Peak Period

Therefore, we must check that an average waiting time during the peak period of 2. 6 x 28. 7 = 77 seconds before a transaction entered into the terminal can be accepted.

In the present application, this will generally be no problem.

For the multi-drop line, a 60% maximum line load is recommended. This means that we have to add the load of the terminals connected to the multi-drop line. (In our example, this is 10 + 13 + 14 = 37%.) As the growth factor is already included in all our figures, we can use the configuration as it is proposed.

### 4. 5. 11 CORE REQUIREMENTS

For a more detailed explanation of the figures used below, we refer to subsection 3. 7.

| | |
|---|---|
| DOS (Supervisor + Transient) | 8K bytes |
| BTAM | 4. 3K bytes |
| Message control | 3. 5K bytes |
| Communications serviceability facilities (incl. checkpoint and restart) | 5K bytes |
| Application program | 10K bytes |
| Total | 30. 8K bytes |
| Safety factor, 20% | 6. 2K bytes |
| | 37. 0K bytes |

Note: For buffering, we have chosen one buffer per line.

Because the terminal speed of the 1050 terminal is low in comparison with the time required to fill the buffer from disk, the flip-flop buffering technique does not have any significant advantage in line and terminal utilization. To combine batch processing with the T/P application described, we need at least a System/360 with 64K. Then we would have: 64 - 37 = 27K available for the batch application programs.

### 4. 5. 12 QUEUING (Refer to Subsection 3. 4)

The results of the relative load calculations for the different devices indicate that the effect of queuing can be disregarded.

We have already calculated that during the peak hour the system accepts 423 messages. This means that the average arrival rate is:

$\lambda = \frac{423}{3600} = 0.12$ transaction/second. The non-weighted average processing time per transaction is:

$\frac{217 + 739 + 1,866}{3 \text{ transaction types}} = 741$ msec. Then, the average service rate is:

$$\mu = \frac{1,000}{741} = 1.4 \text{ transactions/second.}$$

The utilization rate is:

$$r = \frac{\lambda}{\mu} = \frac{0.12}{1.4} = 0.1$$

On the basis of the previous calculations, we can derive the following figures for exponential distribution:

| | | | |
|---|---|---|---|
| Queue Size | Mean = | 0. 1 | |
| | Percentile 90 | 0. 5 | |
| | Percentile 95 | 0. 7 | |
| Queuing Time | Mean = | 1. 2 x 741 ms | = 890 ms |
| | Percentile 90 | 2. 8 x 741 ms | = 2,075 ms |
| | Percentile 95 | 3. 6 x 741 ms | = 2,670 ms |

The conclusion is that the mean increase of the internal response time is 149 msec. As there are no special response time requirements, the effect of the results calculated above can be disregarded.

### 4. 5. 13 RELIABILITY REQUIREMENTS (Refer to Subsections 3. 10 and 3. 11)

There are no special requirements for this system. As we have one terminal per warehouse, we have to include suitable emergency procedures.

A solution would be to write the invoices for rush orders by hand in the event of terminal or computer breakdown.

If the computer is not down, inquiries can be made by telephone or telex. Stock updating can always be done after the terminal and/or computer have been repaired.

## SECTION 5.

## 2260 SYSTEM DESIGN

The IBM 2260 Display Station, which operates in conjunction with the IBM 2848 Display Control Unit, uses a kind of TV screen to display characters. These characters may be typed in on an attached keyboard in order to be checked before transmission to the computer, or may be sent from the computer as an instruction or an answer.

## 5. 1 2260 DISPLAY STATION

This unit consists of:

an optional cathode ray tube to display information;

an optional numeric or alphanumeric keyboard to enter data.

The basic display station consists only of a screen. The keyboard is a feature. 64 different characters can be displayed:

26 alphabetical
10 numerical
25 special symbols (includes space and new line)
3 control symbols (cursor, check and start manual input)

The information is displayed within an area of 22. 86 x 10. 16 cm.

The 2260 may be located 2, 000 feet from the 2848 Control Unit.

## 5. 2 2848 CONTROL UNIT (Figure 140)

The 2848 Control Unit provides:

A common buffer: which acts as the central point for the transfer of data between the interface and the 2848, and for the transfer of data within the 2848.

A character generator: which produces the video signals for the characters.

Display adapters: which contain a delay-line buffer storage and associated control logic to service two 2260 Display Stations. Video data bits are placed in the delay line serially and are continuously displayed and regenerated until erased or replaced by other data.

Printer Adapter: which contains a buffer storage and the circuitry required to control an IBM 1053, Model 4 Printer.

Interface: for connecting local or remote 2260's to the System/360.

Three models of the 2848 are available, with characteristics shown in Figure 141.

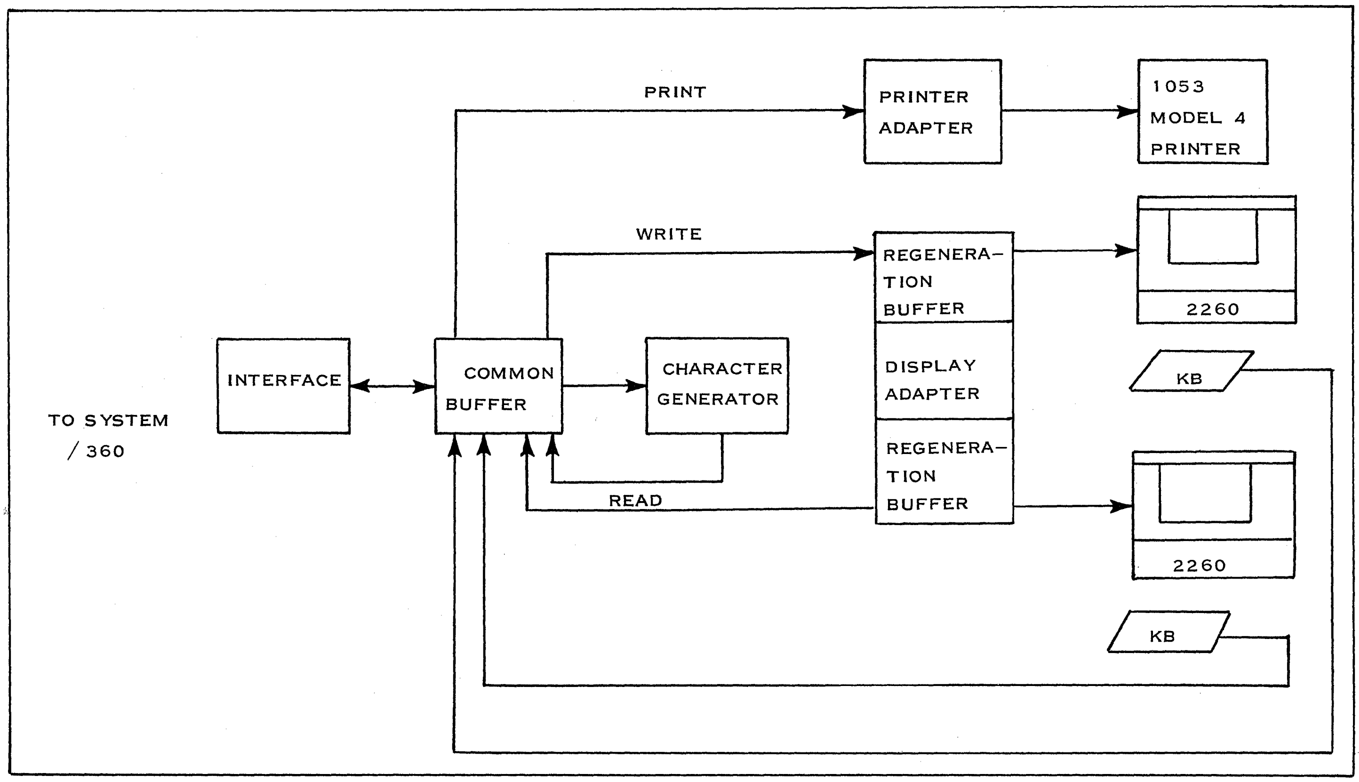


Figure 140. Data Flow Diagram

| 2828 MODEL NUMBER | MAXIMUM NO. OF CHARACTERS DISPLAYED | NO. OF LINES PER DISPLAY | CHARACTERS PER LINE | MAXIMUM NO. OF 2260'S SERVED PER 2848 |
|---|---|---|---|---|
| 1 | 240 | 6 | 40 | 24 |
| 2 | 480 | 12 | 40 | 16 |
| 3 | 960 | 12 | 80 | 8 |

Figure 141. Characteristics of the IBM 2848

The connection between the computer and the 2848 (see Figure 142) can be:

Local: The control unit is attached directly to the multiplexor-selector channel of the System/360 and may be located up to 100 cable feet from the processor. The 2260's can be located up to 2,000 feet from the 2848.

Remote: Communication with the System/360 is done via the 2701 Transmission Control Unit and a communication line.

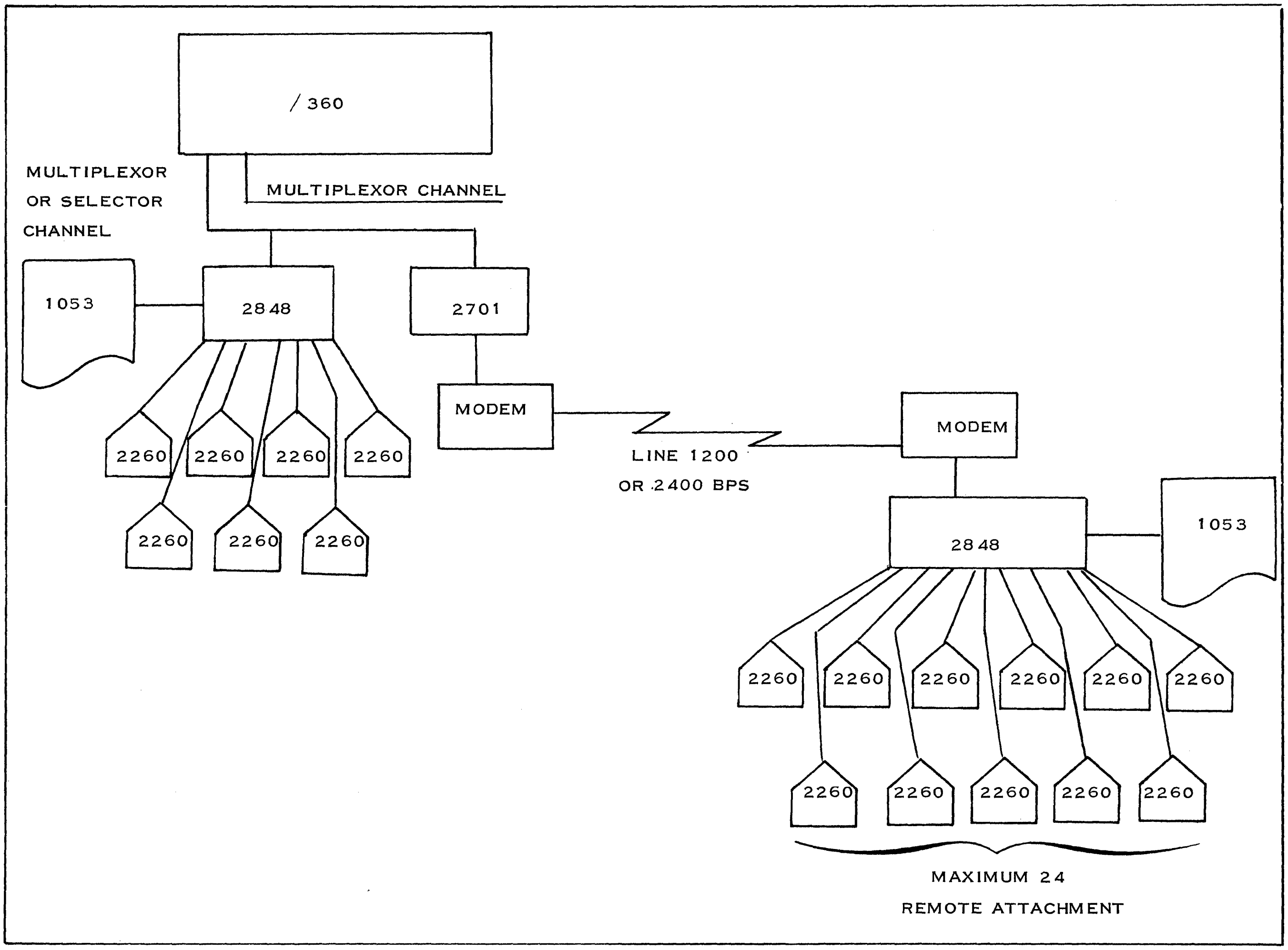


Figure 142. Connection Between the Computer and the IBM 2848

The information keyed in is displayed on the screen: a visual check is possible. To send messages to the computer, the operator has available the following symbols:

Cursor — May be either destructive ▬ or non-destructive ▮ . Denotes the position that the next character entered will occupy.

Start MI — ◣ This symbol indicates the beginning of data that is to be transferred from the 2848 buffer. It appears on the screen by depressing the start key or by transmission from the channel initiated by the computer program. In the latter case, the computer determines when the operator can key in a message. After a successful message transfer, the cursor remains in its previous position and the Start MI symbol is deleted.

NL — ◥ prohibits the transfer of all data displayed between the NL symbol and the end of the display line containing the symbol when the 2260 is polled by the CPU. The new line key on the 2260 places the NL symbol in the cursor position. The cursor moves to the first display position of the next lower display line.

Check — ■ indicates the detection of an error either during the transfer of data between the channel and the 2848 buffer or caused by pressing a key with no associated character code.

In the event of error, the operator can modify a part of a message. For example, the backspace key permits the operator to backspace the cursor to the display position in error and make the correction. The destructive cursor, when backspaced into a display position containing a character, erases the character. Backspacing the non-destructive cursor does not cause displayed data to be erased.

The erase display key, when operated in the upshift condition, erases the entire display and places the cursor in the first displayable position.

The start key operated with a Start MI symbol displayed on the 2260 screen and the shift key depressed erase all data displayed between the Start MI symbol and the cursor, and places the cursor in the position following the start symbol. This does not erase data displayed between an NL symbol and the end of the display line containing that symbol.

Figure 143 illustrates a displayed message format.

| L | CODE | QTY | U. P. | VALUE |
|---|---|---|---|---|
| 1 | 2,315 | 17 | 2.00 | 34.00 |
| 2 | 2,570 | 22 | 0.50 | 11.00 |
| 3 | 3,723 | 5 | 3.00 | 15.00 |
| 4 | 4,567 | 3 | 7.00 | 21.00 |

Figure 143. Displayed Message Format

We can use one of the following methods to change the QTY 5 to 4 in line 3 of the message in Figure 143.

a. Directly replace the digit 5 by 4, if the non-destructive cursor is installed.

b. Type a new line, below the one displayed, such as:

   3 QTY 4

   which is usually simpler for the operator and for the programmer.

For further information concerning the use of different keys and symbols, see IBM 2260 Display and IBM 2848 Display Control, Form A27-2700.

As an example, we list the operations necessary to send a message from a 2260 Display Station.

1. Position the cursor in the first position in which it is desired to begin the message.

2. Depress the start key. The Start MI symbol appears on the screen in the positions previously occupied by the cursor. The cursor is advanced to the next display position.

3. Key in the message. The cursor is advancing automatically.

4. Depress the enter key. This locks the keyboard and signals to the channel that a

message is waiting for transfer in the 2848 buffer. If necessary, the keyboard can be unlocked by using the erase key at the expense of erasing the displayed characters.

5. Upon completion of the transfer of the message to the channel after a Read DSMI command was issued, the keyboard is restored and the MI symbol deleted. This signals to the operator that the message was successfully transferred. A Short Read DSMI command causes the message to be transferred, and after the transfer the keyboard is restored, but the Start MI symbol is not deleted.

## 5. 4 CODES AND ADDRESSING TECHNIQUES

### 2260 LOCAL

The code used by the 2848 attached directly to a System/360 is the Extended Binary Coded Decimal Interchange Code. This code provides 256 characters, of which 64 may be used in data exchanges between the channel and the 2260-2848.

The usual technique of addressing input/output units of a System/360 is also used for the 2848 display control and attached devices. The 2848 is addressed by distinct configurations of the four high-order bits of the address byte; the four low-order bits address the 2260 display stations and 1053 printer. If more than 16 devices are attached to the 2848, two address bytes are used. Every time an enter key is depressed on a 2260 keyboard, the attention bit in the status byte of the 2848 is set. This information is handled by the program.

The 2848 DC can be attached to shared-path subchannels, which can handle up to 16 units each. For more than 16 units (2260 DS and 1053 printer) on the same 2848, two shared-path subchannels are required. If attached to unshared subchannels, the 2848 DC requires as many subchannels as connected units, up to a maximum of 25.

### 2260 REMOTE

The code used for the transmission over telephone lines is the American Standard Code for Information Interchange (ASCII).

Between the channel and the 2848, 71 ASCII characters are used. This code is made up of 7 data bits. A parity bit is added to check the transmission, and a start and stop bit to achieve synchronization. One character can be represented in the following diagram:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| St. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | VRC Bit | Stop Bit |

(All the bits have the same duration.)

Figure 144. Representation of One Character (ASCII)

The data exchanged in parallel between the 2701 and the channel is transmitted in ASCII 8 Code. (An additional bit is added between bits 5 and 6 of the basic ASCII code.) That transformation is made in order to make the data compatible with the 8-bit structure of the System/360 byte. The modification is performed in the 2701 unit.

The addressing and polling technique explained in the section on the 1050 System is also used to control the message flow over the telephone line for the 2848-2260 display complex.

This subject is explained more fully in subsection 5. 6 on Addressing and Command Operations for Remote 2260's.

## 5. 5 COMMANDS

Each operation to be performed by the 2848 display complex is determined by a command byte.

For the local application, seven basic commands are available to be issued by the channel to the 2848. The four low-order bits of a command byte define the basic operation. The four high-order bits are used to expand (modify) an operation.

For the remote application, eight commands using the ASCII 8-bit structure are available. The transfer of each command between the channel and the 2848 display control is controlled by ASCII communication control characters.

This is explained more fully in subsection 5. 6 on Addressing and Command Operations for Remote 2260's.

In Figure 145, we show the different available commands and their functions.

| COMMANDS | | FUNCTION EXECUTED AT SELECTED DISPLAY STATION |
|---|---|---|
| LOCAL | REMOTE | |
| READ MANUAL INPUT<br>SHORT READ MACRO INSTRUCTION | SPECIFIC POLL – 2260<br>SPECIFIC POLL – PRINTER<br>GENERAL POLL | TO TRANSFER A MANUALLY ENTERED MESSAGE TO THE CHANNEL |
| READ FULL DS BUFFER | | TO TRANSFER ALL DATA DISPLAYED TO CHANNEL |
| WRITE DS BUFFER STORAGE | | TO TRANSFER DATA FROM CHANNEL TO DISPLAY |
| WRITE 1053 BUFFER STORAGE | | TO TRANSFER DATA FROM CHANNEL TO PRINTER |
| LINE ADDRESS WRITE | | TO SELECT A DISPLAY LINE WHEN WRITE OPERATION STARTS (LINE ADDRESS FEATURE REQUIRED) |
| ERASE DS BUFFER STORAGE | | TO ERASE ALL CHARACTERS DISPLAYED ON SCREEN |
| | ERASE/ WRITE ADDRESSED DS | TO ERASE ALL CHARACTERS DISPLAYED AND ASSURING THAT THE MESSAGE IS WRITTEN ON SCREEN |
| NO OPERATION | | TO CAUSE THE 2848 TO ANSWER WITH THE NORMAL ENDING SEQUENCE |

Figure 145. Commands and Their Functions

Note:

If data is to be transferred to the channel by a read manual input, short read manual input, specific poll or general poll command, the Start MI symbol must be displayed on the screen before the data is to be transferred. Upon completion of any of the above operations, with the exception of the short read manual input, the Start MI symbol is deleted from the 2260 screen. Deletion of this symbol signifies to the 2260 operator that the data has been transferred from the 2260 buffer and that another message may be entered.

The difference between the general poll and the specific poll commands is that, with the general poll command, each 2260 display station is tested for manually entered messages pending transfer to the channel. With the specific poll command, only one selected 2260 station is tested for a manually entered message pending transfer to the channel.

Use of the general poll command gives a higher effective speed because fewer communication control characters are needed for a certain amount of data characters (reduces line utilization) than with the specific poll command. This will be explained in the next two subsections.

However, with the general poll, it is necessary to unblock the messages by the program. This requires more core.

When the short read manual input command (available only for local operation) is used, it is different from the read manual input command in that the Start MI symbol is not erased. This saves 50 ms and therefore gives a proportionally higher throughput.

Provision must be made in the user program to write over an undesired Start MI symbol or erase the screen prior to the entry of a new input message. The improvement of throughput performance will be considered in greater detail in subsection 5. 10 on Data Entry Type of Operation.

## 5.6 ADDRESSING AND COMMAND OPERATIONS FOR REMOTE 2260's

The following ASCII communication control characters are used to establish and maintain an orderly flow of traffic over the communication line:

STX (Start of Text) — Precedes a sequence of characters which forms a block.

ETX (End of Text) — Analog to End-of-Block character.

CAN (Cancel) — Indicates a transmission error. This character is transmitted only by the 2848.

ACK (Acknowledge) — Transmitted by receiver as affirmative answer to the sender.

NAK (Negative Acknowledge) — Transmitted by receiver as a negative answer to the sender.

SOH (Start of Heading) — Beginning of address (for write command).

EOT (End of Transmission) — To indicate the conclusion of a transmission and used instead of SOH, as the first character of the addressing sequences in the specific poll, general poll, and read operations.

LF (Line Feed) — Format effector used as part of text. Used as a carriage return when the line is not completely displayed.

Every command issued by the channel to a remote terminal is transmitted in a definite format called "addressing sequence". This sequence is formed of four bytes:

| Byte No. | Description |
|---|---|
| 1 | SOH or EOT, places the 2848 in control mode |
| 2 | address of the 2848 |
| 3 | address of the 2260 |
| 4 | specifies the command to be executed. |

Only the principal communication control sequences will be explained here. A more detailed explanation can be found in 2260 Display Station and 2848 Display Control, Form A27-2700.

SPECIFIC POLL TO A 2260 DISPLAY STATION (Figure 146)

The specific poll tests for the presence of a manually entered message awaiting transfer at a selected 2260 to the channel. If a message is pending, the specific poll command causes the message to be transmitted to the channel. If a message is not pending, the 2848 responds with an EOT.

The 2848 receiving the specific poll command tests the following conditions at the selected 2260: enter key depressed and start symbol displayed.

If both conditions are present, the 2848 transmits

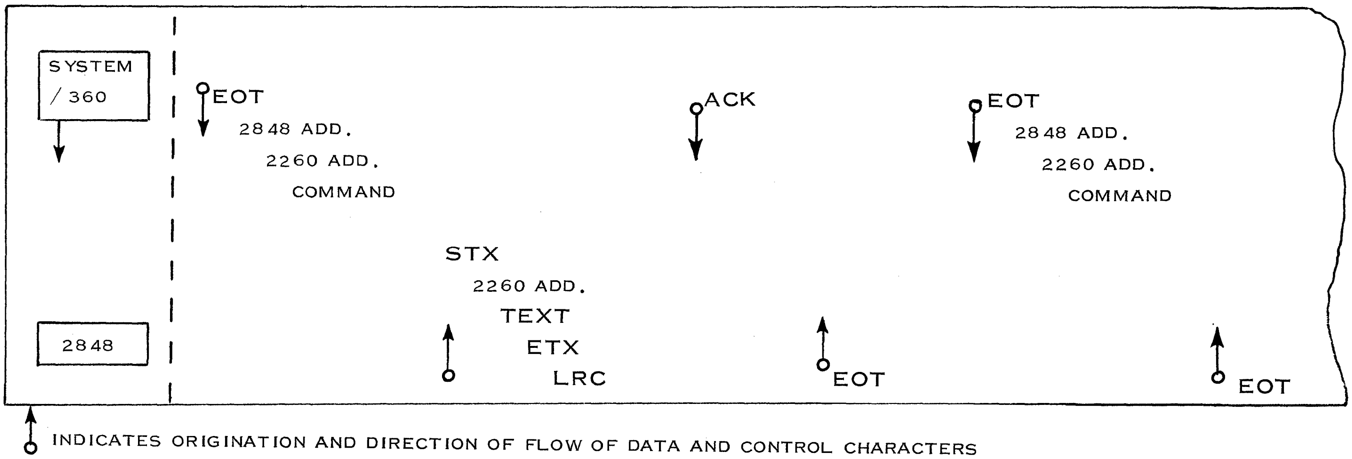


Figure 146. Specific Poll to a 2260 Display Station

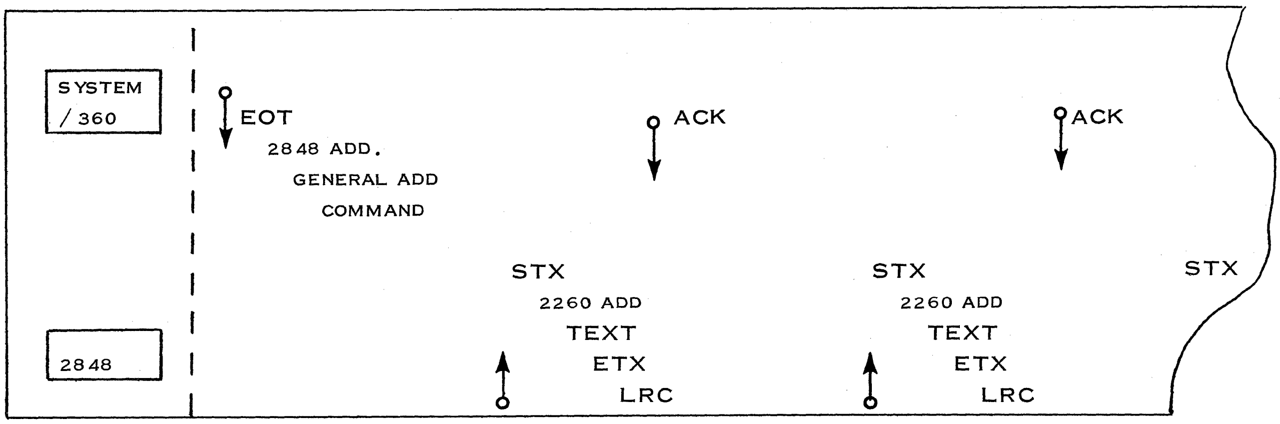


Figure 147. General Poll

start of text, address of the selected 2260, all the data displayed between the start symbol and the cursor, ETX and LRC. The channel responds with ACK (which causes the start symbol to be erased from the screen and the keyboard to be restored). The 2848 responds to ACK with EOT and the operation is concluded.

GENERAL POLL (Figure 147)

The messages of all 2260's, where the MI symbol is present and the enter key is depressed, are transmitted serially to the CPU.

If the printer feature is installed, upon receipt of the general poll sequence the 2848 first tests the printer request condition. If printer request is not set, polling of the 2260's is performed. If there are no messages to be sent, the 2848 responds with EOT.

Upon completion of the message transfer, the channel responds with ACK and the general poll continues until all 2260's have been polled and pending messages transferred. When the last one is sent, the 2848 sends EOT and the operation is concluded.

ERASE/WRITE ADDRESSED DS (Figure 148)

This command is an erase command combined with a write sequence. It causes the 2848 to erase the message displayed on the screen of the selected 2260 DS and causes the cursor to be positioned at the upper left of the screen. When only the addressing sequence is used (not followed by write sequence), it is employed to erase the screen of a particular display station.

The buffer is filled as if it were any 2260 operation. If a line is not completely filled with data characters, the LF special character causes the next character to start the new line without sending blank characters.

WRITE DS LINE ADDRESS (Figure 149)

This command permits the selection of a particular line on the display at the beginning of a message. The cursor is positioned in the first displayable position of the specified line. This makes it possible to keep both the inquiry and the response displayed.

READ ADDRESS FULL DS BUFFER (Figure 150)

This command causes all character data stored in the buffer of the selected 2260 DS to be transferred to the channel. Data is transferred starting at the

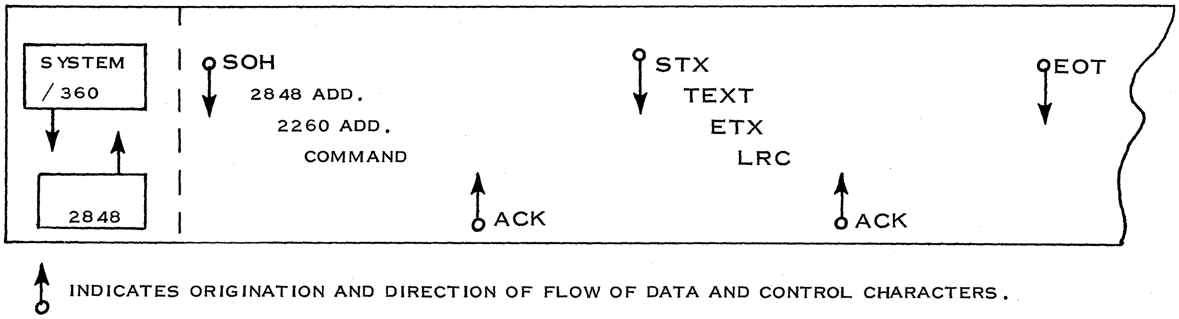
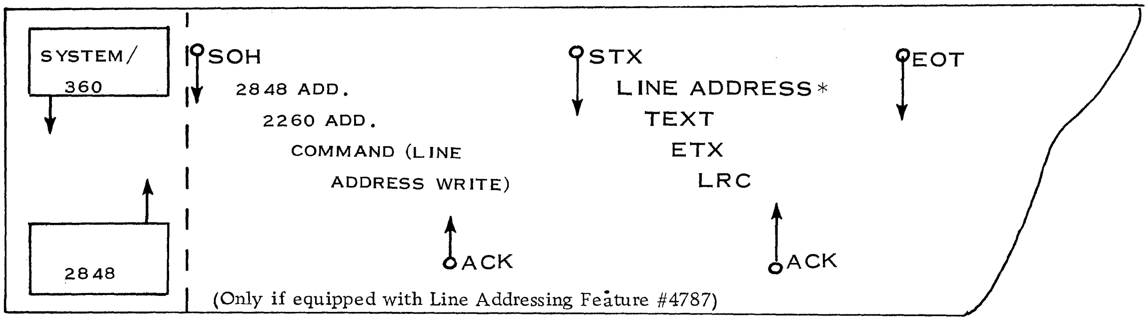

SYSTEM /360
2848
SOH
2848 ADD.
2260 ADD.
COMMAND
ACK
STX
TEXT
ETX
LRC
ACK
EOT
INDICATES ORIGINATION AND DIRECTION OF FLOW OF DATA AND CONTROL CHARACTERS.

Figure 148. Erase/Write Addressed DS


SYSTEM/ 360
2848
SOH
2848 ADD.
2260 ADD.
COMMAND (LINE ADDRESS WRITE)
ACK
STX
LINE ADDRESS*
TEXT
ETX
LRC
ACK
EOT
(Only if equipped with Line Addressing Feature #4787)

*This is the starting line code.

Figure 149. Write DS Line Address

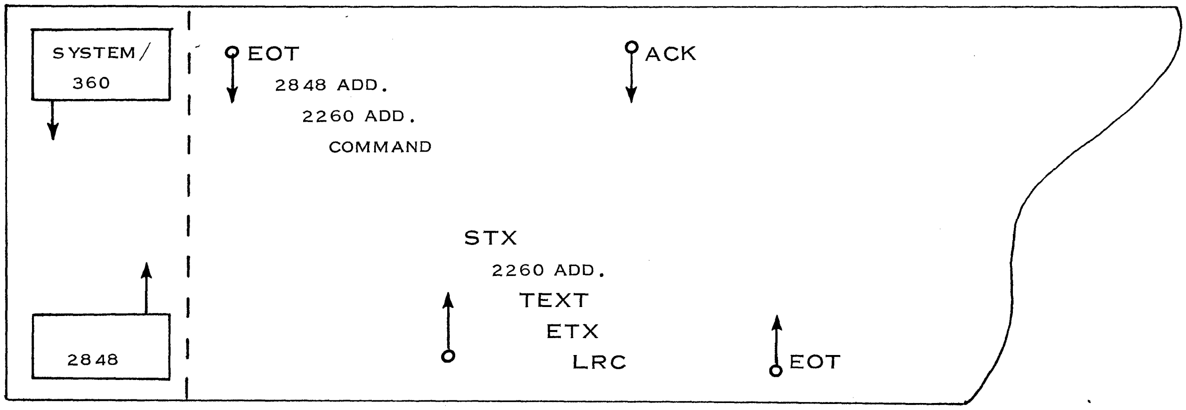
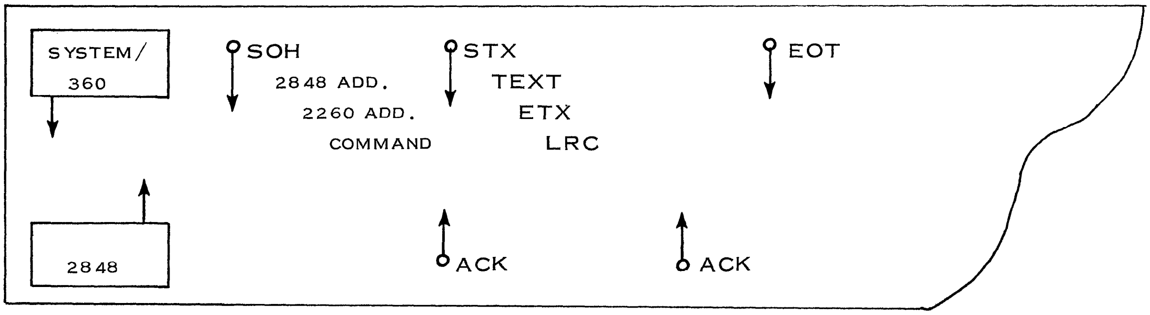


Figure 150. Read Addressed Full DS Buffer



INDICATES ORIGINATION AND DIRECTION OF FLOW OF DATA AND CONTROL CHARACTERS.

Figure 151. Write Addressed 2260 Display Station or Printer

upper left of the screen and ending at the lower right corner. Included among the characters transferred, if they are displayed, are check symbol, start symbol, and destructive cursor. The non-destructive cursor is not transferred.

WRITE ADDRESSED 2260 DISPLAY STATION (Figure 151)

This command is used to transfer data from the channel to the screen of the selected 2260 display station or to the printer.

## 5.7 CALCULATION OF EFFECTIVE LINE SPEED FOR REMOTE OPERATION

Not only the transmission of the data characters must be taken into consideration, but also the turnaround times of the modems or the adapters, and the transmission of control characters.

This is the reason for the decrease in the real speed of the line. In fact, depending on the commands used, the effective speed is between 80 and 100 characters per second, for at least a block of 100 characters, instead of 120 characters per second (which is the theoretical transmission speed at 1200 bps).

The method of computing the effective time taken to transmit a message is as follows:

1. For a 3977 Modem, the turnaround time for full-duplex operation is about 8.5 ms. Include this turnaround time only when the 2848 receives its address from the 2701.

2. Add the time required to transmit the control characters (address, LRC, etc.).

3. Add the time necessary to transmit the text of the message itself, including LF character at the end of the lines, if needed. Refer to Figures 152 and 153 for the timings of the 2701 and 2848. These figures show the segments that make up the setting up of a transmission between the 2701 and the 2848.

It is very difficult to give a general formula for computing the time required for transmission of a message because this time is essentially dependent on the command which is executed. To illustrate the above, we have chosen a general poll as input command and an erase/write DS as an output command.

For input, the timing is:

| | |
|---|---|
| Time to transmit one character at 1200 bps | $\frac{1000}{120} = 8.4$ ms |
| Number of text characters | X |
| Time for the first message | 1 turnaround time for DC address of 8.5 ms plus 8 control characters plus 50.4 ms |
| Times for N other 2260's with a message | N x 67 ms plus N x 5 control characters. |
| End control | 2 control characters plus 70.1 ms |

For half-duplex mode over 4-wire, the maximum time becomes:

8.5 ms + (8 x 8.4) + (X x 8.4) + 50.4 ms +

N x 67 ms + N (5 x 8.4 + X x 8.4) + 2 x 8.4 +

70.1 ms = 129 ms + 10 x 8.4 + (N + 1) (X x 8.4)

+ N x 5 x 8.4

= 213 ms + 8.4 N (5 x X) x 8.4X

= 213 ms + 8.4 [N (5 + X) + X] (Maximum time)

= 162.9 ms + 8.4 [N (5 + X) + X] (Minimum time)

If we take the same numeric values, the time to poll 5 display stations, each with a 10-character message, becomes 0.8 seconds maximum and 0.75 seconds minimum.

Effective line speed: 62 characters/second

If a specific poll were used, it would have taken (N + 1) [221.4 + 8.4 X] (maximum time) = 5 [221.4 + 8.4] = 1.5 seconds.

For output (erase/write DS), the timing is:

| | |
|---|---|
| Turnaround time 2848 Address | 8.5 ms |
| Control character time | 10 characters x 8.4 ms |
| No. of text characters | Y |
| No. of output messages | M |
| 2828 ACK response | 2 x 20 ms |
| Total time = M [132.5 + 8.4Y] | |

If we take the same five display stations, but with a 100-character message to be sent on each, the time is:

| | |
|---|---|
| Time: | 4.86 seconds |
| Effective speed: | 103 characters/ second |

Conclusion:

The blocks of data characters should be as long as possible in order to achieve the highest line speed and general poll should be used as often as possible.

| 2701 SEGMENT | TIME | 2848 SEGMENT | TIME |
|---|---|---|---|
| LINE DELAY | ROUND–TRIP TELEPHONE DELAY BASED ON DISTANCE | | |
| PROGRAM DELAY AND CHANNEL DELAY | TIME FOR PROGRAM TO RESPOND TO AN ANSWER FROM THE 2701 | | |
| EOT | 8.4 MS FOR 120 CPS | | |
| PROGRAM DELAY AND CHANNEL DELAY | TIME IT TAKES CPU TO PLACE NEXT CHARACTER IN 2701 | | |
| DC ADR* | 8.4 MS FOR 120 CPS PLUS 8.5 MS | | |
| PROGRAM DELAY | TIME IT TAKES CPU TO PLACE NEXT CHARACTER IN 2701 | | |
| DS ADR | 8.4 MS FOR 120 CPS | | |
| PROGRAM DELAY AND CHANNEL DELAY | TIME IT TAKES CPU TO PLACE NEXT CHARACTER IN 2701 | | |
| FUNCTION (GENERAL POLL) | 8.4 MS FOR 120 CPS | | |
| | | LINE DELAY | ROUND–TRIP TELEPHONE DELAY |
| | | 2848 POLLING DELAY ** | 20 MS |
| | | EOT | 8.4 MS |

* When the 2848 recognizes the DC ADR, the subset is turned on. A 3977 that is strapped down requires 8.5 ms before it transfers the clear to send signal to the 2848.

** The 2848 polling delay occurs after the subset has finished its turn-on delay. This polling delay is 20 ms if the answerback is going to be EOT.

Figure 152. Negative Response to General Poll

| 2701 SEGMENT | TIME | 2848 SEGMENT | TIME |
|---|---|---|---|
| LINE DELAY | | | |
| PROGRAM DELAY | | | |
| EOT | | | |
| PROGRAM DELAY | SAME AS FIGURE 152 | | |
| DC ADR | | | |
| PROGRAM DELAY | | | |
| DS ADR | | | |
| PROGRAM DELAY | | | |
| FUNCTION (GENERAL POLL) | | | |
| | | LINE DELAY | ROUND-TRIP TELEPHONE DELAY |
| | | STX * | 17 MS POLLING DELAY PLUS 33.4 MS |
| | | DEVICE ADR | 8.4 MS FOR 120 CPS |
| | | TEXT | N X 8.4 MS |
| | | ETX | 8.4 MS FOR 120 CPS |
| | | LCR | 8.4 MS FOR 120 CPS |
| LINE DELAY | ROUND-TRIP TELEPHONE DELAY | | |
| PROGRAM DELAY AND CHANNEL DELAY | PROGRAM DELAY AND CHANNEL DELAY | | |
| STX | ACK 8.4 MS | | |
| EOT | | LINE DELAY | |
| | | EOT 8.4 MS ** | STX 8.4 MS *** |

* The STX will be delayed for 17 ms after the subset has finished its turn-on delay. There is an additional 16.7 to 33.4 ms delay from transmission of STX before first character of the text is transmitted.

** There is a delay of 36.7 to 70.1 ms before EOT is transmitted

*** There is a delay of 16.7 to 67 ms before STX is transmitted.

Figure 153. Positive Response to General Poll

## 5.8 LINE AND DEVICE LOAD

From subsection 3.4 on Queuing and Response Time, we learned that for T/P applications practically each required facility involved in the whole process should be designed so that its load is much less than 100%.

### LINE LOAD

The display complex (2848 - 2260's) can technically be considered as a multi-drop connection (see Figure 154). At any given time, only one display adapter (which is the buffer for a 2260 display station) can send or receive information via the telephone line, but during this time the other 2260 keyboards can operate and messages can be inputted to the 2848. If we have one 2848 per telephone line, we can have a maximum of 24 display stations connected to one line. However, technically, it is possible to attach more 2848's to one telephone line. So we must consider the traffic of all 2260's connected to a line for every load calculation.

The above considerations mean that we may expect an increase in response time as a function of the line load. For example, with an exponential distribution and a 50% line load, we derive that the mean waiting time before a message is transmitted, due to queuing, is $\frac{0.5}{1 - 0.5} = 1$ times the transmission time. With a 60% line load, this delay is $\frac{0.6}{1 - 0.6} = 1.5$ times the transmission time for a message.

### DEVICE LOAD

#### 2848 Display Control

The load of a 2848 in a remote operation has already been determined by the line load which can be accepted.

In a local operation, the internal speed of the 2848 is practically equal to the byte transfer rate between the channel and the display control unit. In this case, the device itself can cause queuing because of the random arrival of the transactions. The considerations previously mentioned for line load concerning the delay before a message is transmitted also apply to the 2848.

Note: In general, we can state that the line and/or 2848 load has to be limited to 50-60%.

#### 2260 Display Station

With the considerations mentioned in the section on Throughput Calculations, we may calculate the average time a display station is occupied to execute one transaction.

The acceptable load per display station during the peak period depends on the acceptable waiting time before a transaction can be keyed in. If $\rho$ is the load of the display station, the waiting time ahead of the terminal will be $\frac{\rho}{1 - \rho}$ times the transaction time.

For a data-entry type of operation, where the display stations replace keypunches, we have to limit ourselves to an average daily load of 50% per station. Data entry is not supported for remote operation.

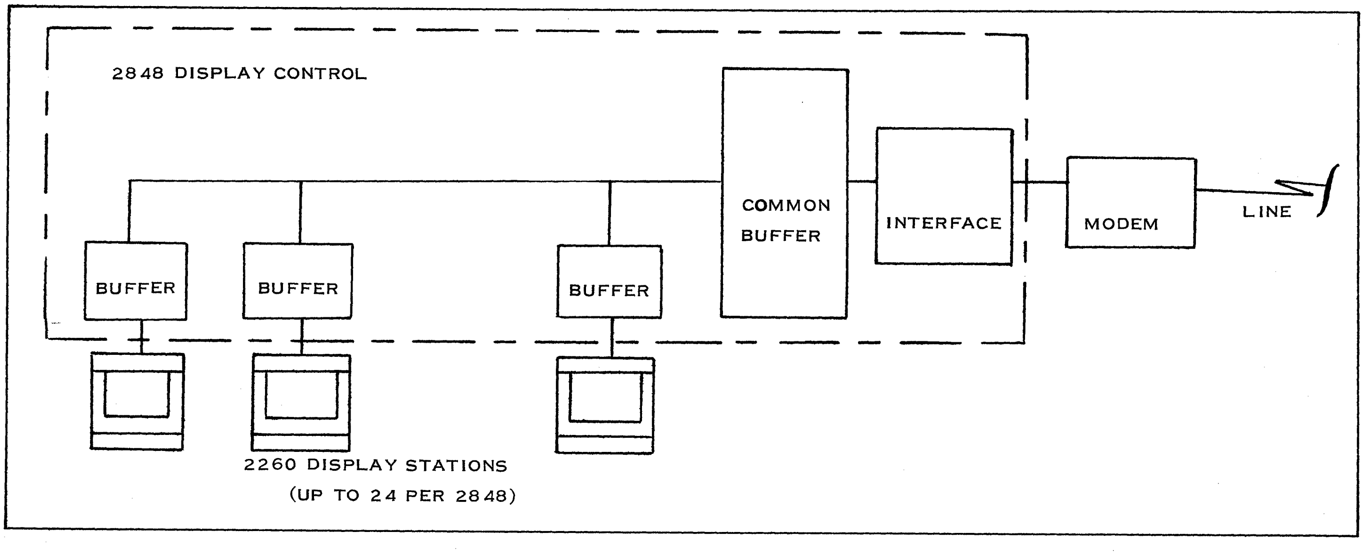


Figure 154. Line Load

## 5. 9 THROUGHPUT CALCULATIONS

The actual throughput is limited by:

Keying rate of the operator, which depends on:

- type of keyboard used (alphameric or numeric)
- time to check a message before it is sent to the computer
- time to read and check the answer displayed.

Technical limitations:

- Transmission speed:

  Local: ~1200 characters per second, which are mainly effective data characters;

  Remote: 120 characters per second. In this case, we have to consider that the effective speed will be slower because we have to take into account the transmission control characters. (This was previously explained in subsections 3. 5 and 5. 7. )

  As the common circuitry operates in serial mode (it can do only one thing at a time), requests for service are queued and handled on a priority basis. If a heavy demand is placed on the 2848 to service messages across the interface, and at the same time operators are keying rapidly in the attached 2260 keyboards, the 2848 may not be able to respond to a keystroke request in time before the next key depression. This subject will be considered in greater detail in Data Entry-Type Operation, subsection 5. 10.

- 2848 Display Control Unit:

  For local use, provision must be made for the following delays:

A. 2260 Enter Sequence

1. 16. 7 ms - Placing of EOM symbol and positioning of cursor

2. 0-16. 7 ms - Synchronizing of channel and 2848

3. Assuming the use of the READ DSMI:

   a. The first line is calculated at 0. 4 ms per character position from position following start symbol to last position on line (40th or 80th).
   b. Subsequent lines (except the last) are calculated at 0. 4 ms per character position from position 1 to position 40 or 80 (either 16 ms or 32 ms will be the result).
   c. The last line is calculated at 0. 4 ms per character from position 1 to the character position immediately preceding the EOM symbol.

4. Following each line, except the last, a pause of approximately 34 ms is experienced.

5. Ending time for the READ DS MI is approximately 50 ms. Ending time for the SHORT READ DS MI is 0. 4 ms.

B. 2260 Write Sequence

1. 0 - 16. 7 ms -- synchronization of channel and 2848.

2. Each line (except the last) is calculated at 0. 4 ms per character position from position 1 to position 40 or 80. The last line is figured from position 1 to the last position written.

3. Following each line, except the last, a pause of approximately 34 ms is experienced.

C. 2260 Input Example

1. Assume:

   a. Start symbol in the first position of the first line;
   b. READ DS MI command used;
   c. 100 consecutive character positions to be read starting with the character immediately following the start symbol;
   d. Based on a 2848 Model 1 with 40 characters per line

displayed on the 2260 and a total of 6 lines.

2. Timings

| | |
|---|---|
| 16.7 ms | Start time |
| 16.7 ms | Synch. Time (worst case) |
| 15.6 ms | Read time - 1st line - .4 ms x 39 char. |
| 34.0 ms | Interleave time |
| 16.0 ms | Read time - 2nd line - 40 char. |
| 34.0 ms | Interleave time |
| 8.4 ms | Read time - 3rd line - .4 ms x 39 char. |
| 50.0 ms | Ending time |
| 91.4 ms | Total |

D. 2260 Output Example

1. Assume

a. 100 consecutive character positions are written
b. Write DS line address command used
c. Based on a 2848 Model 1 with 40 characters per line displayed on the 2260 and a total of 6 lines

2. Timings

| | |
|---|---|
| 16.7 ms | Synch. time |
| 16.0 ms | Write time - 1st line - 40 char. |
| 34.0 ms | Interleave time |
| 16.0 ms | Write time - 2nd line - 40 char. |
| 34.0 ms | Interleave time |
| 8.0 ms | Write time - 3rd line - .4 ms x 20 char. |
| 124.7 ms | Total |

E. Total Time Input Plus Output

| | |
|---|---|
| 91.4 | Input |
| 124.7 | Output |
| 216.1 ms | |

The total time per transaction for the 2848 DC is 216.1 ms. If we accept a 50% load for this unit, it will then allow $0.5 \times \frac{60}{0.216} = 140$ transactions per minute.

This means that with a maximum of 24 equally loaded display stations per 2848 DC, there would be $\frac{140}{24} = 5.9$ transactions per minute, per display station. This rate of 10 cps is superior to what a human operator can handle and there is no problem concerning the load of the 2848 DC. The only point we have to bear in mind is the overall keying rate.

- Remote 2260

To calculate the time to transmit one input and one outout message, we used the formulas derived in subsection 5.7, Calculation of the Effective Line Speed for Remote Operation for the specific poll and the erase/write command, respectively.

- Input (2848 ⟶ System/360)

Message length is:

X = 100 data characters

$\text{Time} = (N + 1) \ [221.4 + 8.4X]$

$= 1,061.4 \text{ ms}$

Note: N = 1

- Output (System/360 ⟶ 2848)

Message length is:

Y = 100 data characters

$\text{Time} = M \ [132.5 + 8.4Y]$

$= 972.5 \text{ ms}$

Note: M = 1

The total time per transaction (in + out) of the 2848 DC is 2,034 ms. If we accept a 50% peak load for this unit, it then allows $0.5 \times \frac{60}{2.0} = 15$ transactions per minute.

This means that with a maximum of 24 stations per 2848 DC, there would be $\frac{15}{24}$ or .625 transactions per minute, per display station.

As a human operator can handle about one transaction per minute, the effective transmission speed between the communication channel and the 2848 DC is the limiting factor.

## 5. 10 DATA ENTRY-TYPE OF OPERATION

The 2260-2848 display complex was originally designed and announced as a data inquiry terminal. However, since its announcement, the 2260 is being used more and more for data entry-type operations.

A. Conventional card punch units can be replaced with the following advantages:

- Reduced turnaround time results in improved total system operation.

- Operator productivity can increase by an average of 35% over conventional keypunches because of:

  Relatively quiet operation of the 2260;

  Fully variable input formats allowed;

  Possibility of error correction during punching.

- Card costs of approximately $22. 00 per month, per card punch station, will be eliminated.

- Card inventories and peripheral document handling will also be eliminated.

- Management control on current work achieved and on operator productivity can be handled by the processor.

B. The second objective is to replace the verifiers, which can be done in either of two ways or a combination of both:

- Rekeying of the document to compare character per character with the first keying. Any detected error is shown on the display.

- Reasonableness checks and verification of data by the processor.

Obviously, the second method is preferable, since it avoids rekeying and allows extensive checks. In fact, control can be immediately performed by the operator on the source document, and errors can immediately be corrected without rekeying the entire record or recycling the documents through the system. This may even eliminate former manual control prior to punching or allow some kind of automatic coding.

C. Processor utilization can be optimized by real-time or partial processing of data entered into the system, thus shortening the elapsed time for total processing.

D. The operator can even interpret the source document according to information retrieved from the files, such as item availability.

For data entry-type of operation, especially where the main objective is to replace keypunches, there are some important advantages in using the local operation:

- Communication speed is increased to approximately 1200 characters per second between the CPU and the 2848 DC.

- An automatic CPU interrupt results from a keyboard request and raises a special "attention signal". Then polling is avoided and response time is shortened.

- The 2701 Transmission Control Unit, data sets and telephone line are not required, which further decreases the price of these low-cost displays and improves system reliability.

In subsection 5. 9 on Throughput Calculations, we have outlined a method of designing the display complex.

However, with data entry-type of operation, we can expect:

- Many 2260 display stations per 2848 control unit;

- High volume of activity per station;

- Burst keying rate of the operators.

With this combination of parameters, the 2848 may not be able to respond to a keystroke request in time to restore the keyboard before the next key depression. When this occurs, the operator may sense a resistance to her finger pressure.

When a key is depressed, the character involved is placed in the common buffer. However, to display it on the screen, the character has to be translated in the character generator and stored in the display adapter.

This results in a request for service from the common circuitry of the 2848. The time required for each keyboard key to perform its function depends on the function involved. This is, for example, for a character and a space of 0.8 ms, a backspace of 16.7 msec, etc.

If there is heavy demand across the interface, it may happen that there is not enough time available to service the keystroke requests in time.

Local 2260's (Refer to Figures 155 to 160)

When transferring messages from the processor to the 2848, there is a pause before the start of each line after the first line of 33.4 msec. This time is partly available for servicing the keyboards.

To establish the recommended maximum number of 2260 display stations for a single 2848, performance curves are available. These curves are plotted as a function of:

- Throughput Per Station (Records/Hour/Station)

  This is the number of records to be entered per hour at each station. Pick a representative peak hour and assume the total job load of the system to be divided equally among all active stations.

- Record Length (Characters)

  This is the average number of positions on the 2260 Display Station between the start symbol of the record to be entered and the cursor.

- Keying Rate (Characters/Second)

  This is the average instantaneous rate of key depressions by the operator when she is productively keying.

- The graphs do not reflect the interleave read of 33.4 ms.

It is recommended, as a guide, that, for an alphanumeric keyboard, the curve for 2.4 characters/second and for a numeric keyboard 3.3 - 5.1 characters/second be used. It must be kept in mind that the speed is also operator-dependent.

The following are two examples showing how performance curves in typical applications are used.

Example 1

A wholesaler receives 900 telephone orders per day (8 hours) from retailers. Each order has, on the average, 10 entries. Twelve operators enter the orders at 2260 stations as they are received on the phone. Each entry consists of a 20-character record, for which there is a one-line response, indicating whether the item called for is in stock.

In addition, there are two low-volume stations for inquiries and corrections. Can one 2848 control unit handle this configuration?

Keying rate for this type of application is assumed to be slow -- 2.4 cps. Also, assume the READ DS MI command is used. Record length is 20 characters. Throughput is calculated as follows:

$$\frac{900 \text{ orders/day} \times 10 \text{ entries/order}}{8 \text{ hours/day} \times 12 \text{ stations}} = 94 \text{ entries/hour/station}$$

Plot the parameters on the graph in Figure 155. The result indicates that the control unit can handle up to 24 x 2260's.

Example 2

Operators enter data from a stack of documents. Each document results in a 60-character record. Output consists of an online response. Operator keying rate is 3.3 cps. Ten seconds are required to check the response and acquire the next document.

If 36 operators are needed, how many 2848 control units are required?

$$\text{Throughput} = \frac{3600 \text{ sec./hr.}}{\frac{60 \text{ char./rec.}}{3.3 \text{ char./sec.}} + 10 \text{ sec.}} = 129 \text{ rec./hr./station}$$

Assuming the READ DS MI command is used, plot the above parameters on the graph in Figure 156.

This indicates that six stations per control unit can be handled and that, for 36 stations, six control units are required.

However, it is probable that the application will not require the use of the READ DS MI command, so that the SHORT READ DS MI command can be used. If this is the case, the graph in Figure 157 shows that 20 stations per control unit can be accommodated. Thus, for 36 stations, two control units are required. The SHORT READ DS MI command is always recommended for data entry.

Remote 2260's

We may expect that the limited remote data entry-type of operation will cause no difficulties for remote 2260's because:

- The total volume of activity and, therefore, the number of display stations per 2848 control unit is already very limited by the effective communication speed between the channel and the 2848 DC. Remote 2260's cannot handle high volume, true data entry because of the communication line-speed limitation.

- The data set interface provides a 35-character delay line buffer. During a write operation, it is expected that 2 to 6 characters will be transferred in succession, with an internal speed of 0.4 msec per character, to the display adapter. The remainder of the difference between internal and transmission speed will be used to service keyboards. Moreover, turnaround times are also available for this purpose.

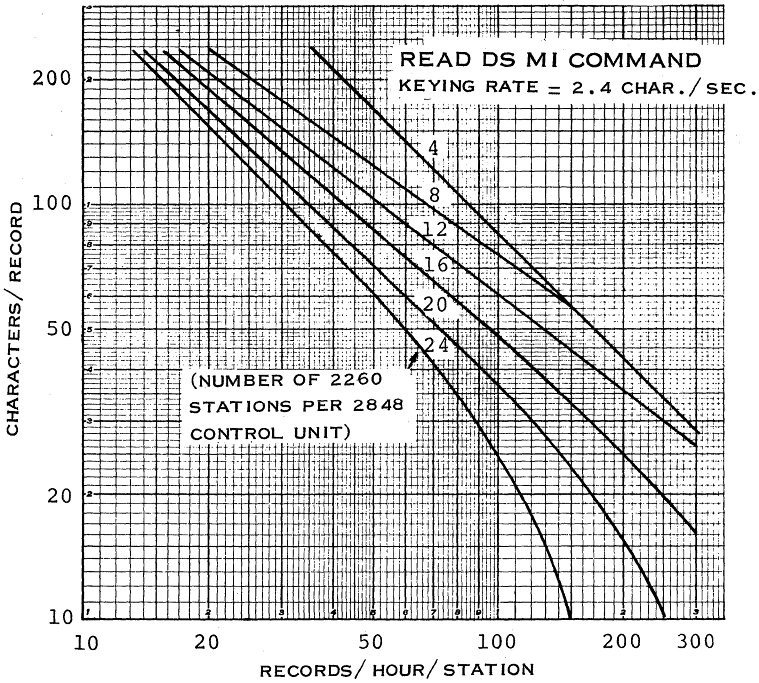


Figure 155. (Graph #1) READ DS MI Command (Keying Rate = 2.4 Characters/Second)

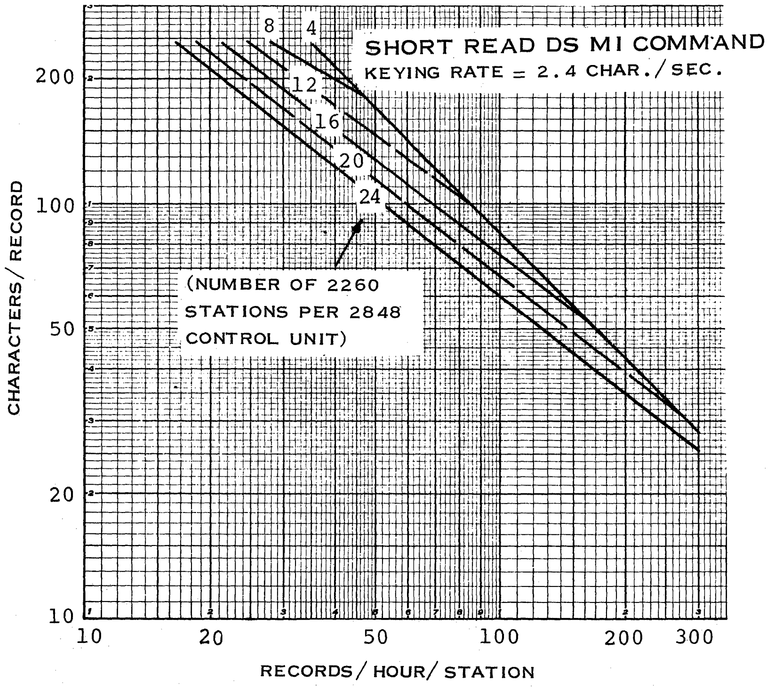


Figure 156. (Graph #2) SHORT READ DS MI Command (Keying Rate = 2.4 Characters/Second)

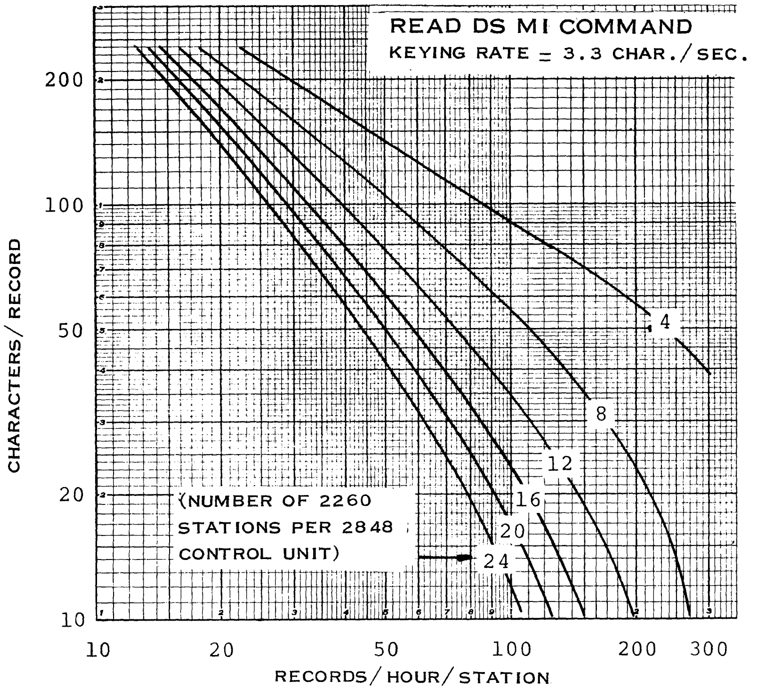


Figure 157. (Graph #3) READ DS MI Command (Keying Rate = 3.3 Characters/Second)

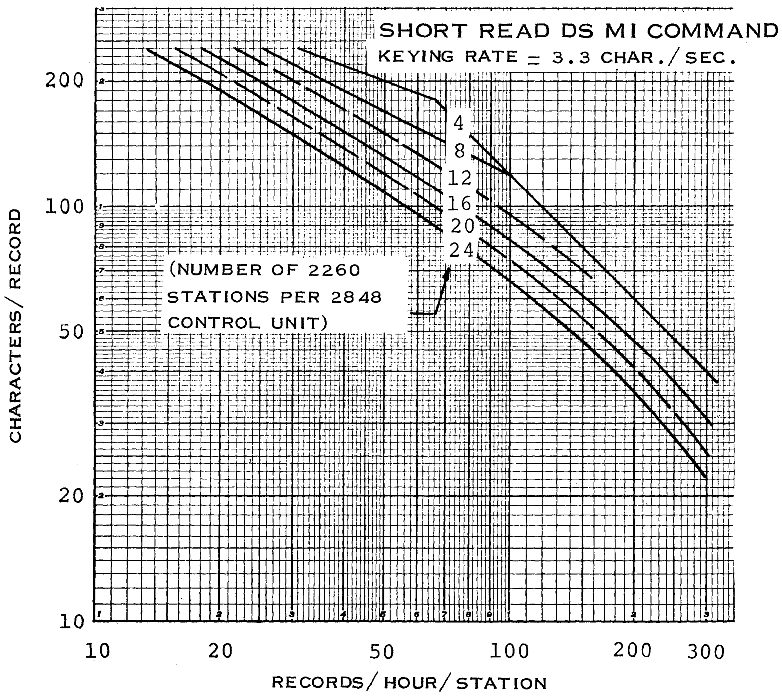


Figure 158. (Graph #4) SHORT READ DS MI Command (Keying Rate = 3.3 Characters/Second)

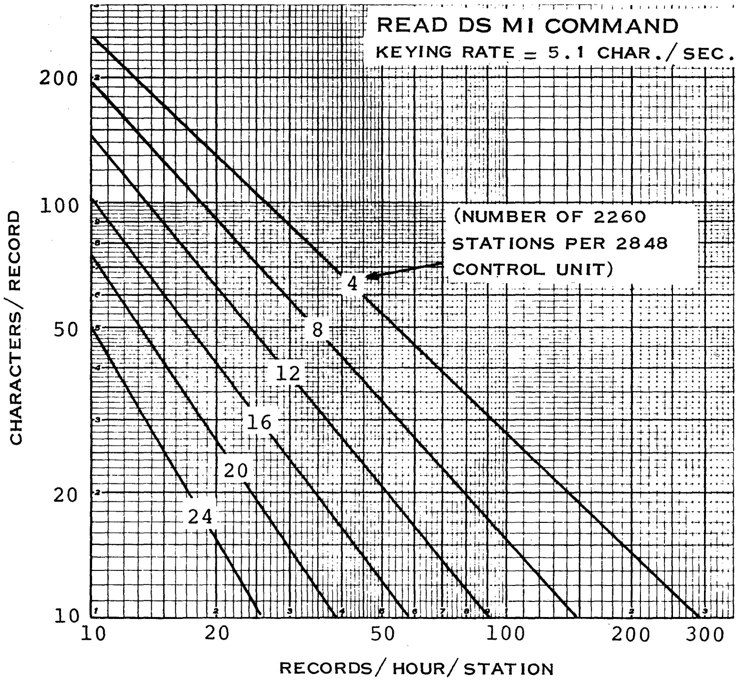


Figure 159. (Graph #5) READ DS MI Command (Keying rate = 5.1 Characters/Second)

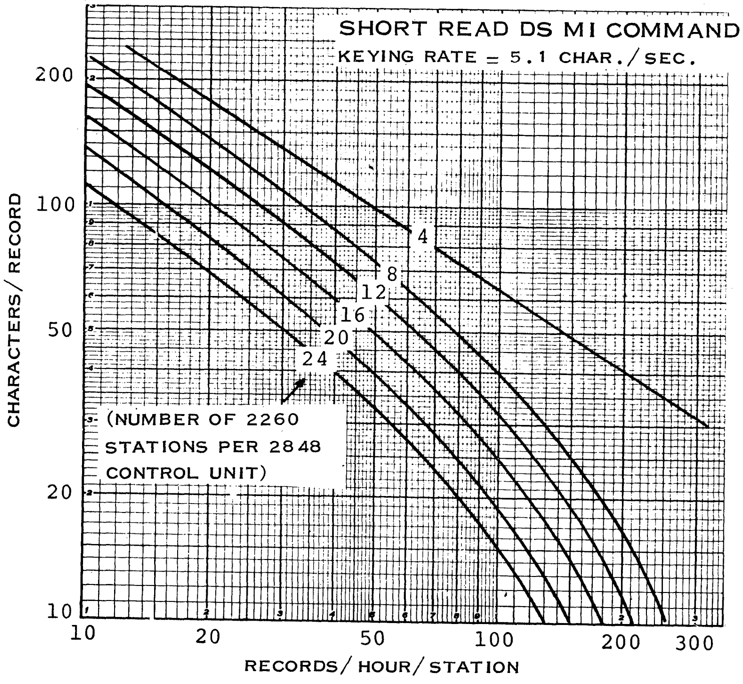


Figure 160. (Graph #6) SHORT READ DS MI Command (Keying Rate = 5.1 Characters/Second)

## 5.11 CHANNEL LOAD AND INTERFERENCE

For further information concerning this subject, the reader is referred to subsection 3.9, Channel Load and Interference and to Average Reduction in Processing Capability Caused by Channel Interference in System/360 Models 30, 40, and 50, Form Z20-1780.

### LOCAL 2260's

The interface adapter of the 2848 DC is designed to operate in the single-byte mode. The 2848 attached to a System/360 channel is a non-overrun device.

Only if the channel is overloaded (70% or 80%) will there be a loss of performance. In that case, which is exceptional for small systems, the 2848 DC can also be attached to a selector channel.

In view of the possible high transfer rate between the channel and the 2848, the multiplexor channel interference on the CPU must be considered.

The CPU (micro program) times for a single-byte transfer on the MPX channel for Models 30, 40, and 50 are 62.25, 28.8, and 12 μs, respectively. Channel interference calculated on these figures would be 6.2, 2.9, and 1.2 times N, respectively. (N is the total mean rate of single-byte requests in kilobytes per second.) (See Figure 161.)

| SYSTEM/360 MODEL | CHANNEL INTERFERENCE IN %CPU UTILIZATION |
|---|---|
| 30 | 5 N |
| 40 | 2.5 N |
| 50 | 1 N |

Figure 161. Channel Interference

If we consider the same inquiry example previously used in subsection 5.9, it provides a practical upper limit of traffic for one 2848 DC. The transfer rate is equal to:

100 + 100 = 200 bytes per transaction and a maximum of
140 transactions/minute, or 2.33 transactions/second.

Then N = 200 x 2.33 = 466 bytes/second ≃ 0.47 kilobytes/second. For a Model 30, the channel interference would be: 0.47 x 5 = 2.4%.

### REMOTE 2260's

In this case, channel load and interference are determined by the 2701 connected to the multiplexor channel.

In subsection 3.9, Channel Load and Interference, Figure 56 shows channel interference as a function of the byte rate for a 2701.

With a 2848-2260 display complex, the maximum speed is 120 bytes/second, or a channel interference of less than 1%.

## 5. 12 ERROR CHECKING AND RECOVERY

The program error routines are activated when error-condition interrupts are tested. These error conditions are indicated for the local 2260's in the sense and status bytes of the 2848, and for remote 2260's in the sense and status bytes of the 2701.

These interrupts are, for example, caused by detecting:

- Invalid commands and/or addresses;

- Parity errors in data transfer:

  for write operations indicated by a displayed check symbol ( ▬ ); and for a poll operation, the keyboard is restored and the MI symbol ( ▶ ) not deleted from the screen;

- Buffer parity errors:

  for a remote application, a special character, CAN, is sent between the message and the ETX symbol;

- A disparity exists between the LRC byte and the accumulated LRC for remote 2260's.

For a more detailed description of this subject, refer to IBM 2260 Display Station and the IBM 2848 Display Control, Form Z27, 2700.

In general, it is recommended that all detectable machine and transmission errors be logged. If an error is detected, at least one retry will be attempted before the operation with the selected device is terminated and the operator notified. This should minimize interruptions due to infrequent transients. It is recommended that the number of errors used as threshold criteria to terminate operation of a display should be easily changeable. An acceptable number will depend upon the application, the system configuration, and the user.

5.13 BUFFERS (Core Requirements)

For 2260 reading, we need buffers equal to the number of displayable positions: 240, 480, or 960, according to the particular model.

Even to read a short message, we need maximum-length buffers to avoid possible difficulties.

Moreover, in DOS Extended, the nature of dynamic buffer allocation is such that if shorter buffer segments are used, the system automatically allocates enough segments to contain the maximum-length message. Also in this case, the header prefix occupies 32 bytes in the first segment and text prefixes, 22 bytes in the following ones. A common buffer pool may be used to save core space.

If S is the segment size, we need n buffers to read a 2260 Model III with:

$$(S - 32) + (n - 1)(S - 22) \simeq 960$$

or

$$n \simeq \frac{970}{S - 22}$$

For instance, with S = 80 bytes per segment, this gives:

n = 17 segments required (1,360 bytes).

In general, we may expect that 2 to 4 buffers per 2848 are required for the queued access method.

## 5. 14 PROGRAMMING SUPPORT

Figure 162 shows the announced support for the 2260 Display Station.

| | Local | Remote |
|---|---|---|
| DOS<br>II | BTAM | BTAM<br>QTAM |
| OS-<br>SPS | Express GPS<br>Basic GPS | BTAM<br>QTAM |

Figure 162. Programming Support for 2260

BTAM/DOS

BTAM/DOS supports the 2260 local and remote display stations. All necessary operations can be performed by BTAM macros. For example, for the local 2260, the access method is specified, since no polling is required; but the supervisor must handle a particular kind of conditional interrupts which are presented to the channel by the status and sense bytes. The READ Initial (TI) macro reads in all characters displayed between the start symbol and the cursor (except the characters to the right of an NL symbol) of any station defined in the associated DTFBT, as soon as the operator has depressed the enter key.

For the remote 2260, the READ Initial (TI) macro initiates:

- a specific poll of a display station, or
- a request for a print status, or
- a general poll of a display control unit, depending on the characters specified in the polling list.

BTAM/SPS

BTAM/SPS supports only the remote 2260. There are no functional differences in the BTAM in DOS. There are only some coding differences.

QTAM/DOS and QTAM/SPS

QTAM/DOS and QTAM/SPS support only the remote 2260. The main advantage of QTAM is that the control program includes message queuing and is generated according to definition statements. (See subsections 2. 6. 3. and 2. 6. 6. )

OS-SPS/Express GPS

OS-SPS/Express GPS stores only the signals when they occur in a special list of the GCB (Graphic Control Blocks) attached to each device and located in the Supervisor area.

Then the user program must periodically check these GCB's for pending signals with a special macro ANALYS; and, if they exist, the program must issue a READ macro to read the related buffer.

OS-SPS/Basic GPS

OS-SPS/Basic GPS effectively interrupts the running program and transfers control to a special Attention Handling Routine (AHR) when the signal occurs. Only the AHR itself may not be interrupted; and if new signals occur during this time they are automatically queued until the end of this routine. Provision for handling input/output errors is also included in both the Express and Basic GPS. The standard error routine analyzes the status and sense bytes for synchronous error conditions and provides appropriate error-recovery procedures for specific error situations.

A more complete description can be found in IBM Operating System/360 Graphic Programming Services for IBM Display Station (Local Attachment) Preliminary Specifications, Form C27-6912, and IBM System/360 Operating System; IBM 2260 Display Station (Local), Form C27-6925.

## 5.15 REMOTE SYSTEM 2260: EXAMPLE

### I. GENERAL COMMENTS ON SPECIFICATIONS FOR THE 2260 INQUIRY AND UPDATING SYSTEM

A. These general specifications are for a customer who has no System/360. We intend to propose to him a system permitting batch processing with some communications capability.

B. The size of this communications network will be up to 2 lines (if load on those lines permits), operating in half-duplex mode over 4 wires, up to a speed of 1200 bps.

C. The system will be designed with 2260 terminals (including their 2848 control unit).

D. Point-to-point leased lines will be considered.

E. The application to be described in the specifications is for a management information system -- a scheduling system for a construction company using PERT.

F. Other types of applications can be included in in the same general procedure described for the application in this example, since each of them can use the 2260 as an information terminal. Some of these applications are:

- manufacturing control
- equipment designing
- automatic seeking of documentation
- airline reservation
- bank and insurance system

The PERT Information System application is given only as an example of how the 2260 is used and can easily be replaced by another information system application.

G. The real-time portion of the application can start as a simple inquiry system.

Inquiry is a function that can be applied to any application without exception. Basically, the inquiry for a real-time system operates as follows:

1. The inquirer wants to know the current status of a certain bit of information.
2. He sends a formatted inquiry message from a terminal to a central computer.
3. The computer recognizes the inquiry type.
4. The computer then goes to the proper files and obtains the proper information.
5. The information is placed in a proper format by the computer.
6. The answer to the inquiry is sent back to the terminal.

H. Updating messages are those that change existing file records. Again, these messages can apply to any application. In this PERT example, the actual updating messages for the schedule records are logged at the time an affirmative message is received from the manager's terminal. The schedule records are lated updated in a batch-processing program.

### II. SPECIFIC PROBLEMS RELATIVE TO THE APPLICATION

As previously mentioned, the customer is a construction company using PERT. This customer wants to perform the following functions:

- During the day he wants all his department supervisors to be in a position to know their work schedule up to the next three months. This inquiry can be made at any time of the day, and the answer delay must be short (6 seconds).
- Each department supervisor must be able to make slight modifications to the schedule. These modifications must be submitted to the general management, who will approve or cancel them.
- If a modification is accepted, this information will be placed in a log file. The file will be used to update the master schedule file after T/P hours.
- Batch-processing of other concurrent applications is required, for example, costs, statistics, payroll, PERT, etc.

Central Computer Requirements

Once a day, the batch application program uses card input to the modified PERT program.

The system has to perform the following operations:

1. Design PERT network for individual jobs.

2. Scheduling.

3. Allocate the resources which, in this case, are the individual departments.

4. Prepare schedules by day, for a three-month period, for each individual department, and formats for output messages.

5. Put information on direct access files, one day per cylinder or group of cylinders; one record per department. Each record may have 480 characters.

6. Duplicate this file, so that the following day it can be used for the online application. The online application uses a duplicate department schedule file while batch-processing updates the master schedule file.

Special Requirements

We assume in our example that the normal use of the inquiry system is as follows:

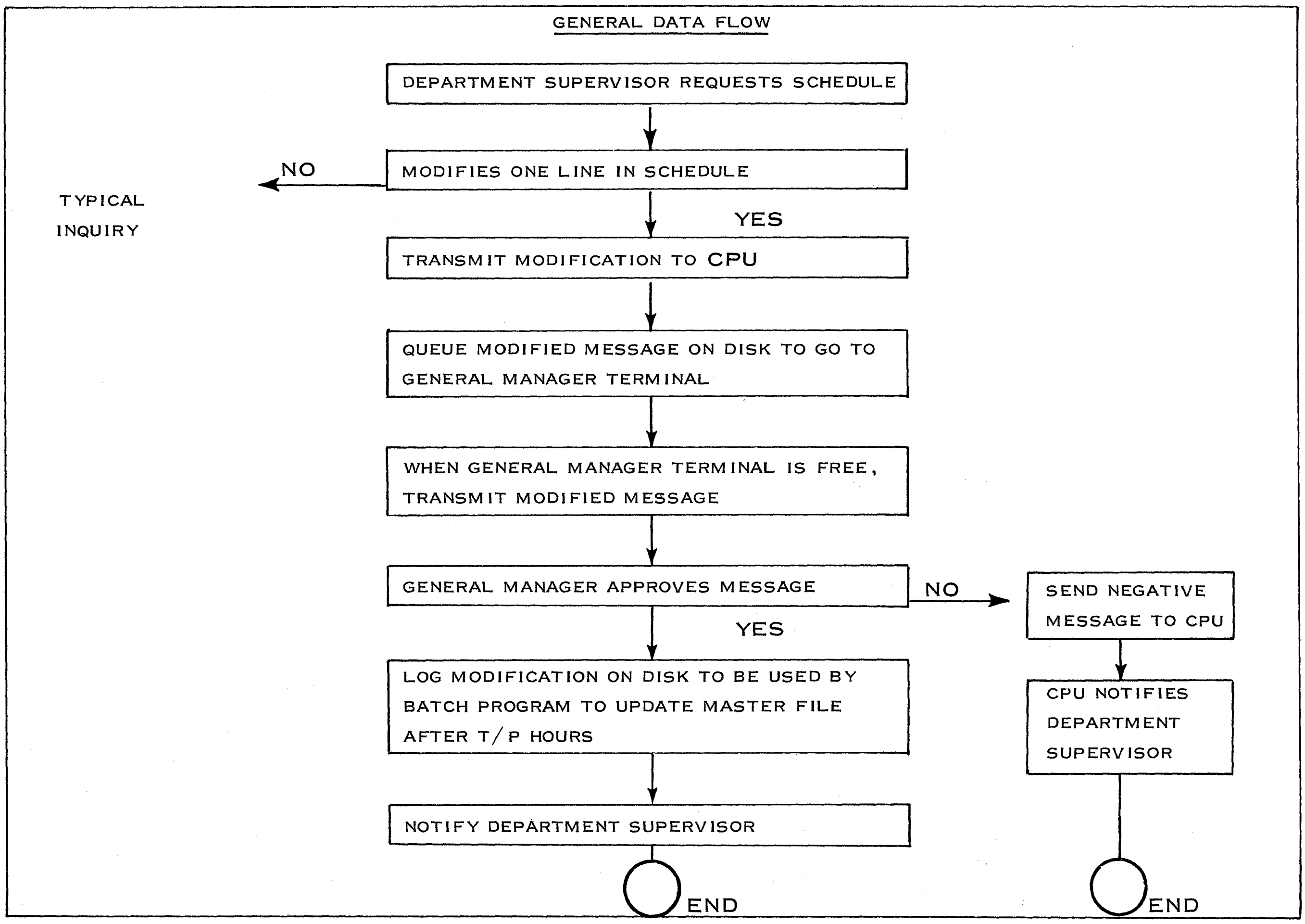
One terminal sends the inquiry after a polling from the CPU. The CPU processes the inquiry and sends the reply message to the requesting terminal. Then it polls the following, and so on. Sometimes a change in the schedule is required, but before being completed it has to be approved by the manager (see Figure 163). The operation is carried out as follows:

- The inquiry of a definite format is sent to the computer. There it is queued and logged on disk.

- When the manager's 2260 is free, the following operations are performed:

  - Read first message from queue

  - Transmit to manager's terminal



GENERAL DATA FLOW
DEPARTMENT SUPERVISOR REQUESTS SCHEDULE
MODIFIES ONE LINE IN SCHEDULE
NO
TYPICAL INQUIRY
YES
TRANSMIT MODIFICATION TO CPU
QUEUE MODIFIED MESSAGE ON DISK TO GO TO GENERAL MANAGER TERMINAL
WHEN GENERAL MANAGER TERMINAL IS FREE, TRANSMIT MODIFIED MESSAGE
GENERAL MANAGER APPROVES MESSAGE
NO
SEND NEGATIVE MESSAGE TO CPU
YES
LOG MODIFICATION ON DISK TO BE USED BY BATCH PROGRAM TO UPDATE MASTER FILE AFTER T/P HOURS
CPU NOTIFIES DEPARTMENT SUPERVISOR
NOTIFY DEPARTMENT SUPERVISOR
END
END

Figure 163. Data Flow: Schedule Modification and Inquiry

- Manager decides affirmatively or negatively

- Transmit to the system

  If YES    update duplicate file

  If NO    send negative answer to the department terminal requesting change

III. CUSTOMER REQUIREMENTS ANALYSIS

Data Flow

The whole operation is summarized in Figure 163 for schedule modification and inquiry.

In order to meet the requirements of this data flow, the next step is to define roughly the terminals that the customer needs.

Terminal Requirements:

The system, in the pre-design phase, will include seven 2260 display stations for the supervisors and one display station for the general manager. Five supervisors' display stations are connected to one telephone line, and the two remaining stations, plus the manager's station, are connected to the other telephone line through the remote 2848's.

Since the supervisors as well as the general manager may make some corrections or additions to a particular schedule, all the display stations must be equipped with a keyboard.

Message Formatting:

From the general data flow, we can see that six different types of message formats are necessary for this application:

| Message Type | Input/Output to CPU |
|---|---|
| (1) Inquiry request | Input |
| (2) Answerback | Output |
| (3) Modification | Input |
| (4) Notification to manager | Output |
| (5) Approval or disapproval message | Input |
| (6) Notification of decision to supervisor | Output |

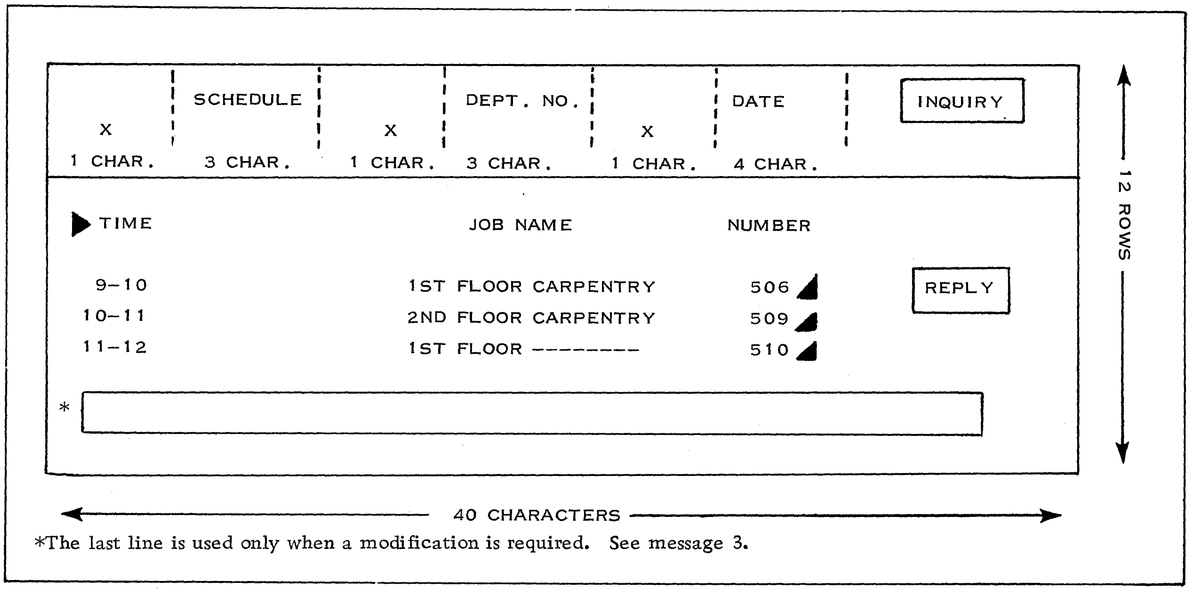
Each of the different message formats has to be defined (see Figure 164):

1. Inquiry Request

   Input message (1) - a total of 13 characters, including spaces

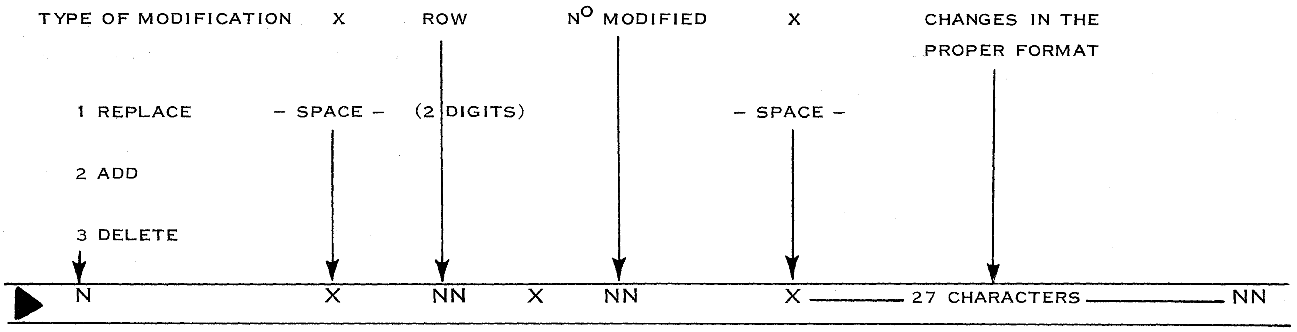
2. Answerback

   Output message (2) -
   20 characters for the first line;
   35 characters per line for the following lines (7 on the average), including NL characters, if any.
   Total - 265 characters.



*The last line is used only when a modification is required. See message 3.

Figure 164. Message Formats: Inquiry and Reply



The total line is composed of 35 characters.
The rest of the displayed message remains unchanged.
Message 3 is composed of 1 line, i. e. 35 characters.

Figure 165. Modification to Message Format

3. Modification to schedule by department supervisor: Only the last row is involved, and the format for that row is shown in Figure 165.

4. Notification of Modification to the Manager:

The CPU creates this message by adding a header and combining the change and modified rows of the input message, as follows:

Dept. No. x Name x Name of Supervisor - Date of Schedule

Type of Modification - second line

Rows - unmodified - schedule line

- modified - schedule line

Since we are not concerned with presenting a complete study, but merely with an example, we give only a general description of the message formats.

The total message length is approximately 160 characters, including NL characters, if any.

5. and 6. Approval/Disapproval, and Notification

The message is the same as the previous one, but the last row is changed.

The manager merely notes approval or cancellation. The whole message is then sent to the supervisor requesting the changes.

This message can be composed of 4 lines (160 characters if the manager decides on some changes).

Traffic Analysis:

Once the message formats are defined, a rough analysis of the traffic is performed to determine the transaction rate during peak hours for all terminals and types of messages. Figure 166 summarizes that analysis.

Figure 167 shows the total daily traffic on the lines.

Total approved messages to update the file are:

25 during peak hour

70 during normal hours (total)

Total: 25 + 70 = 95 messages, which will be logged on disk for further updating of the day's schedule.

IV. SYSTEM DESIGN

A. <u>Peak Line Load Calculations</u> (see subsection 5. 7)

Line 1 - Message 1 (Input)

Average time for sending an inquiry (general poll)

Number of characters: 13

67 ms + (5 x 8. 4) + (13 x 8. 4) = 151. 2 ms

= . 15 seconds

| TERMINAL | | TYPE OF MESSAGE | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | PEAK HOUR | | | | | NORMAL HOUR | | | | |
| | | 1 & 2 | 3 | 4 | 5 | 6 | 1 & 2 | 3 | 4 | 5 | 6 |
| MGR., TERM. A | LINE 2 | ▨ | ▨ | 35 | 35 | ▨ | ▨ | ▨ | 12 | 12 | ▨ |
| 1 SUPER. 1 | | 30 | 5 | ▨ | ▨ | 5 | 10 | 1 | ▨ | ▨ | 1 |
| 2 SUPER. 2 | | 35 | 3 | ▨ | ▨ | 3 | 15 | 2 | ▨ | ▨ | 2 |
| 3 SUPER. 3 | LINE 1 | 40 | 8 | ▨ | ▨ | 8 | 15 | 2 | ▨ | ▨ | 2 |
| 4 SUPER. 4 | | 20 | 5 | ▨ | ▨ | 5 | 8 | 1 | ▨ | ▨ | 1 |
| 5 SUPER. 5 | | 50 | 6 | ▨ | ▨ | 6 | 20 | 3 | ▨ | ▨ | 3 |
| 6 SUPER. 6 | | 40 | 5 | ▨ | ▨ | 5 | 10 | 2 | ▨ | ▨ | 2 |
| 7 SUPER. 7 | | 25 | 3 | ▨ | ▨ | 3 | 15 | 1 | ▨ | ▨ | 1 |
| TOTAL | | 240 | 35 | 35 | 35 | 35 | | | | | |

Note: Each input message produces an output - total of 190 inputs and 190 outputs.

Figure 166. Traffic Analysis Table (Messages/Hour)

| | MESSAGE TYPE | | | | |
|---|---|---|---|---|---|
| | 1 & 2 | 3 | 4 | 5 | 6 |
| LINE 1 | 651 | 90 | ▨ | ▨ | 90 |
| LINE 2 | 240 | 29 | 119 | 119 | 29 |

Figure 167. Total Daily Traffic on Lines

Message 2 (Output)

Time for sending the reply (write DS Line Address)
Number of characters: 265
132. 5 + (8. 4 x 265) = 2, 226 ms
= 2. 22 sec.
Time for one transaction: 2. 37 seconds.
As the number of messages during the peak hour is 175, the total time is 2. 37 x 175 = 415 seconds.

Message 3 (Input 35 characters)

Time for sending the message (general poll): 403 ms.
As the number of messages during the peak hour is 27, the total time is . 403 x 27 = 10. 8 seconds.

Messages 4 and 5: None on that line.

Message 6 (Output 160 characters)

Time for sending the message (erase and write): 1. 34 seconds.
As the number of messages is 27, the total time is 27 x 1. 34 = 36. 2 seconds.

Total transmission time on Line 1:

415 + 10. 8 + 36. 2 = 462 seconds.

Load of Line 1: $\frac{462 \times 100}{3600} = 12.8\%$

Line 2 - Messages 1 and 2

Number of transactions during peak hour: 65
Time for sending messages 1 and 2: 154 seconds

Message 3

Number of messages during peak hour: 8
Time for sending message: 3.22 seconds

Messages 4 and 6

Messages 4 and 6 have the same length and both are output messages.
Number of messages 4 and 6 during peak hour: 43
Time for sending messages 4 and 6: 57.5 seconds

Message 5 (Input 160 Characters)

Time for sending one message: 67 ms + (5 x 8.4) + (160 x 8.4) = 1.46 seconds
Number of messages during peak hour: 35
Total time for sending message 5 is: 51.1 seconds

Total time on Line 2: 266 seconds

Load of Line 2: $\frac{266 \times 100}{3600} = 7.4\%$

B. Selection of Hardware

The total configuration for this application (for the front end) is shown in Figure 168.

C. File Organization

We refer to subsection 3.5 of the Handbook, but will take a short example to illustrate file organization, which is as follows:

1. Department Schedule File:

   We must estimate the volume of the master file where we intend to store the schedule. The disk file is a 2311. The file is organized as follows:

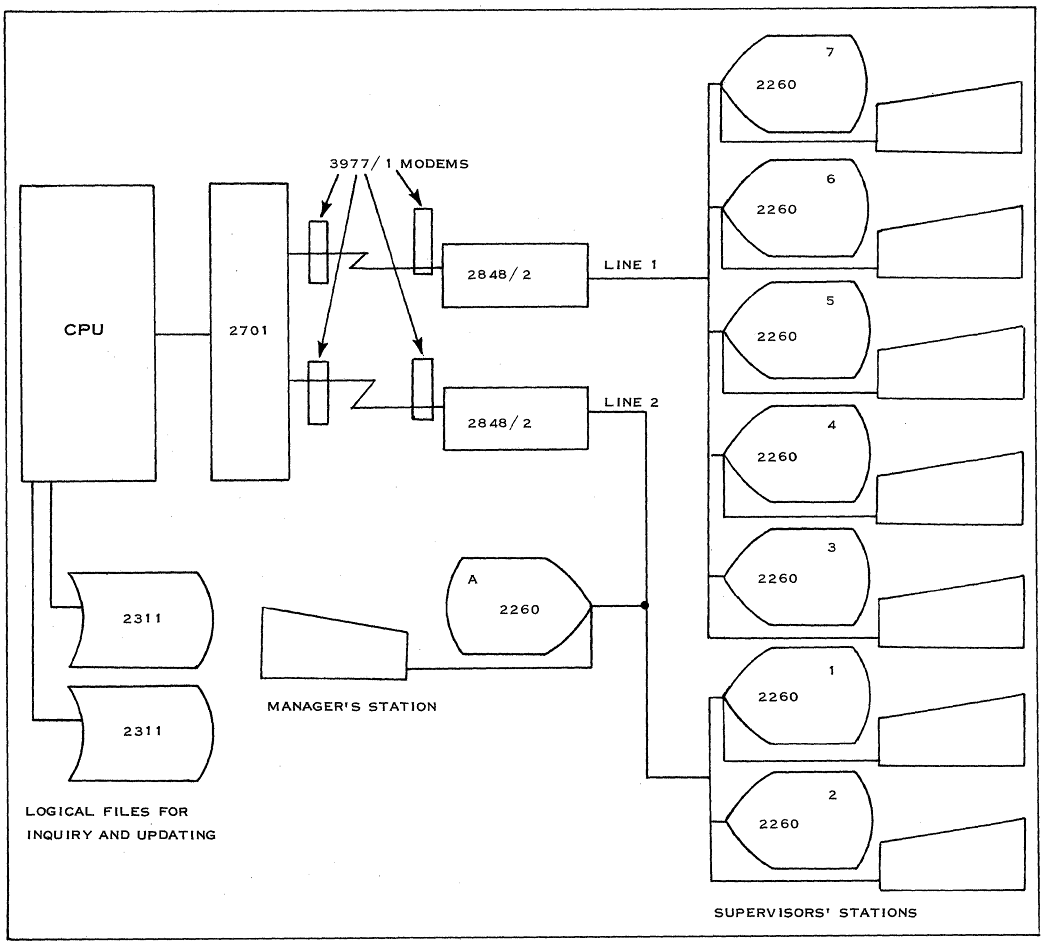


Figure 168. Hardware for 2260 Application

- one day of schedules by department per cylinder or group of cylinders;

- one record per department. There are 7 departments. Each record has 480 characters.

The method used is the direct access (BDAM) We will put the schedule for three months (91 days) in the department schedule file.

The number of bytes per record is 480, and we have 6 tracks per cylinder (from the 2311 chart, X20-1705).

We have 60 records of 480 characters per cylinder; and, as we have 7 departments a day, we can put 60/7 = 8 days per cylinder.

Since a three-month file must be stored (91 days), we need 91/8 = 13 cylinders for the complete file.

2. Log File for Modified Messages:

From the traffic analysis, we found that in one day 25 messages requiring modification were logged during the peak hour and 10 per hour were logged during the normal hours. So the total number of logged messages per day was: 25 + 7 x 10 = 95 messages per day.

Since the format of modified messages is 160 characters per record and we can put 140 records on one cylinder, the log file will need one cylinder.

3. Queue File for Message Switching:

The notification by the manager to the supervisors is the return of the transmitted message: in other words, a small message-switching application. To achieve it, we need a queue file of one message of 160 characters per department, or one cylinder.

Our total file estimate comes to:
13 + 1 + 1 = 15 cylinders.

D. Error Procedure and Recovery

Refer to subsections 3.10 and 3.11.

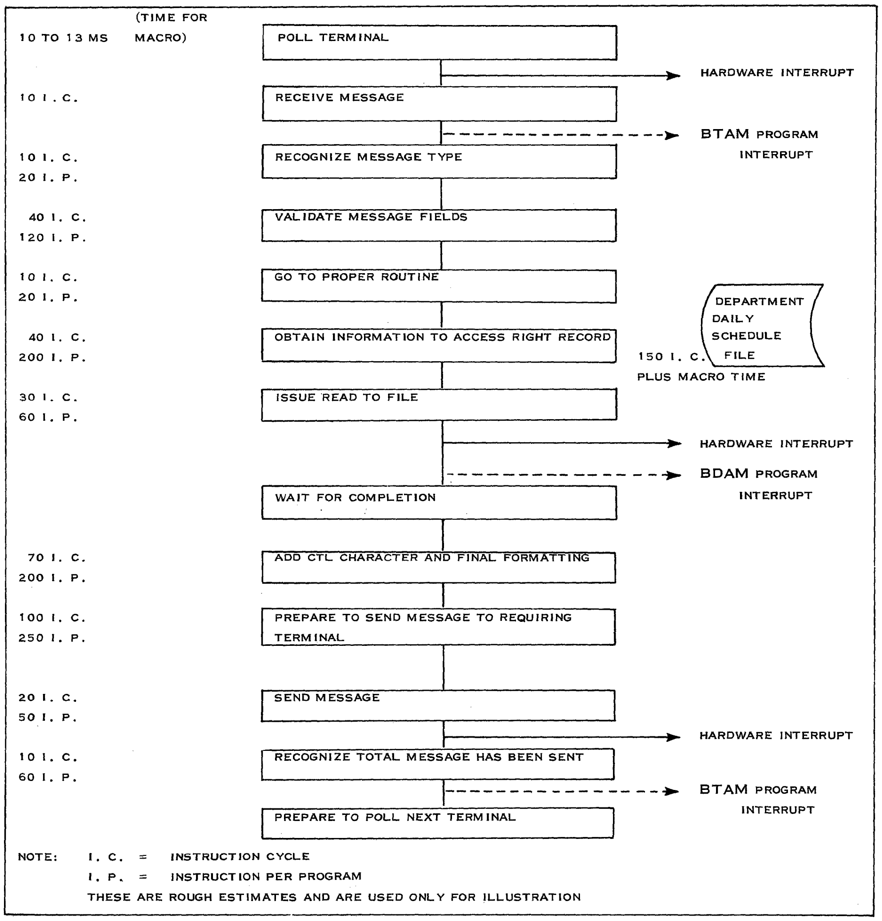


Figure 169. Message Flow

| | |
|---|---|
| • SUPERVISOR (SPS) | 25K BYTES |
| • BTAM (NO DYNAMIC BUFFERING) | 3K BYTES |
| • T/P APPLICATION PROGRAM | 4.275K BYTES |
| (THIS FIGURE COMES FROM THE FLOWCHART IN FIGURE 169: 950 INSTRUCTIONS AT 4.5 BYTES PER INSTRUCTION.) | |
| • BDAM INTERFACE ROUTINE FOR QUEUING AND LOGGING | 0.5K BYTES |
| • SERVICEABILITY AND ERROR RECOVERY | 5K BYTES |
| • BATCH – APPLICATION PROGRAM | 16K BYTES |
| • LINKAGE EDITOR | 16K BYTES |
| • QISAM | 6.7K BYTES |
| • OVERLAY SUPERVISOR | 2K BYTES |
| • PERT * | 100K BYTES |

*Again, it is emphasized here that PERT is used only as an example. The batch application program may be of any size. It could be only 16K, for instance, and this would reduce the core required to 128K.

| | | |
|---|---|---|
| The total core estimate is: | 178. 5K bytes | |
| We add a safety factor of 20%: | 178. 5 | + 35. 5 = 214K bytes |

This result leads us to use a 256K-byte machine. At least a System/360 Model 40 must be chosen

Figure 170. Core Estimates

E. CPU and Channel Load: Total Processing Time

Message Flow:

In order to calculate the time it takes to process a transaction, a flowchart must be developed to show the major processing functions. Figure 169 is an example of such a flowchart.

F. Core Estimates (Refer to 3. 7. 4)

To estimate the core necessary for this application, we have to know how much is required by the supervisor, higher priority tasks such as BTAM, T/P application program, etc., and lower priority tasks. These are illustrated in Figure 170.

G. Processing Time Calculation (Refer to 3. 12)

A calculation of processing time is shown in Figure 171.

H. Response Time

As previously defined, response time is the interval between the time the enter key is depressed and the last character (EOT) of the reply is received by the display station.

Response time is equal to the sum of:

1. Time necessary to send the inquiry
2. Processing time inside the CPU
3. Queuing time (total)
4. Time necessary to send back the reply

1. TIME TAKEN BY BTAM PROGRAM TO READ, CHECK AND ANALYZE THE MESSAGE BEFORE PROCESSING.

THE INPUT MESSAGE IS 13 CHARACTERS AND THE OUTPUT IS 265. A SYSTEM/360, MODEL 40 TAKES ABOUT 120 MS FOR INPUT AND OUTPUT MESSAGE AND LINE CONTROL, FOR A TOTAL OF 278 CHARACTERS. THE INPUT MESSAGE CONTROL WOULD BE 4 MS PLUS 4 MS FOR A BTAM READ.

| INPUT MESSAGE CONTROL | 2311 | CHANNEL | CPU | PROCESSING TIME |
|---|---|---|---|---|
| | | | 8 MS | 8 MS |

| 2. READ RECORD FROM DISK - BDAM | 2311 | CHANNEL | CPU | PROCESSING TIME |
|---|---|---|---|---|
| PROCESS TIME - READ | 10.6 MS | 10.6 MS | 10.6 MS | 10.6 MS |
| SEEK TIME | 75 | | | 150 MS |
| ROTATIONAL DELAY | 12.5 | 12.5 | | |
| RECORD TIME 265 X $7.10^{-3}$ | 1.85 | 1.85 | | |
| | 99.95 | 24.95 | 10.6 | 160.6 |

3. APPLICATION PROGRAM: WE ASSUME 340 INSTRUCTIONS, TAKING 20 MS EACH. THE TIME FOR THE APPLICATION PROGRAM IS 6.8 MS.

4. SEND MESSAGE TO THE LINE: SEND MESSAGES TO THE LINE TAKES 116 MS FOR MESSAGE AND LINE CONTROL PLUS 4 MS FOR A BTAM WRITE.

| | 2311 | CHANNEL | CPU | PROCESSING TIME |
|---|---|---|---|---|
| OUTPUT MESSAGE CONTROL | | | 120 MS | 120 MS |

TOTAL PROCESSING TIME FOR THE INQUIRY AND REPLY IS:

8 MS + 160.6 MS + 6.8 MS + 120 MS = 295.4 MS

Note: All the above figures include a safety factor of 20%

Figure 171. Processing Time Calculation

For our example, we know:

- Time necessary to send an inquiry: .15 seconds
- Time necessary to send the answer back: 2.2 seconds
- Processing time: .30 seconds

Queuing time may be computed by using the formulas in subsection 3.5

As we computed a line load of about 13% for the most heavily loaded line, since the messages are constant the formula $\frac{h \text{ x } (2- )}{2\ (1 - \rho)} = R$ can apply.

$$h = \text{input time} + \text{processing} + \text{output time}$$
$$= .15 + .3 + 2.2 = 2.65$$

$$R = \frac{2.65\ (2 - .13)}{2\ (1 - .13)} = 2.85 \text{ seconds}$$

Since the customer requires a maximum of 6 seconds, the system meets his requirements.

Conclusion: A System/360 Model 40 with 256K-byte storage meets the customer's requirements and may be proposed for this application.

| | DUE TO BATCH | DUE TO T/P |
|---|---|---|
| CPU | 45.0 % | .73% |
| DISK | 25.0% | .60 % |
| CHANNEL SELECTOR | 35.0% | .13 % |
| CHANNEL MPX | 7.0% | ≃0 % |

Figure 172. Utilization Table

Load Calculation

In our example, we assumed there were 190 inputs for the peak hour and 190 outputs. We will derive the loads for the disk, channel, and CPU accordingly. It is assumed here that all message types have an h equal to or greater than the inquiry.

$$\text{Disk Load: } \frac{109.95 \text{ x } 190 \text{ x } 100 \text{ x } 10^{-3}}{3,600} = .6\%$$

$$\text{Channel Load: } \frac{24.95 \text{ x } 190 \text{ x } 100 \text{ x } 10^{-3}}{3,600} = .13\%$$

$$\text{CPU Load: } \frac{138.6 \text{ x } 190 \text{ x } 100 \text{ x } 10^{-3}}{3,600} = .73\%$$

This shows how the calculation has been made. For the whole system, we can make the assumptions (which are purely hypothetical) shown in Figure 172.

Note: The assumptions for utilization are shown here only to indicate to the user that he must calculate the utilization of the units due to batch operation.

CONCLUSION

We see from Figure 172 that, as the limit of 60% is not exceeded, this system is not overloaded.

## 5.16 EXAMPLES

### 5.16.1 SALES ORDER ENTRY WITH 2260

While the unique possibilities of the 2260 as an inquiry unit do not have to be explained, the 2260 is not restricted to this application. More and more it is being used for data entry. To illustrate this flexibility, an example has been chosen which is similar to the 1050 System example. The use of the 2260 for a central sales order entry application will be studied.

Let us assume that the customer has a central warehouse, from which all orders are shipped. The orders arrive mostly by mail in the central location, but provision must be made for rush orders telephoned in by branch offices or directly by customers and for inquiries made over the telephone concerning availability of stock. However, the rush orders and inquiries have a low volume.

Instead of keypunching and verifying, 2260 display stations are used to enter the orders directly into the System/360. The operators key in the customer number, item number, and quantity of all positions of the order. The system immediately:

a. Checks for possible errors (keying or wrong item numbers);

b. Checks the availability of the quantity;

c. Checks possible credit limitations, etc.;

d. Updates the quantity of the item;

e. Debits the customer.

The system then adds to the input information the customer name, item description, and line number. In the second last line on the display, it shows the results of the above checks in the form of messages to the operator. The operator corrects the erroneous positions, and a clean invoice is transferred to disk until the invoices are printed in a batch mode operation.

This method differs somewhat from more conversational methods (which are also possible). According to this method, each line entered would be answered by the CPU immediately without updating. Only the checking would be performed. After the operator has checked the answer, the line is entered and updating is performed. This method is more convenient for the operator. In addition, error correction is easier, but has the disadvantage that the availability of items is no longer valid after the operator has checked the item. Any other station could in the meantime have changed the quantity. Therefore, the first method was chosen.

In the first part, a "local" 2260 system will be studied. Here the 2848 is connected to a system channel. The operators are located nearby. In the second part, we will study what, if any effect, there would be on the performance of the system if the 2260's are installed "remotely". Here we will assume that the 2260's are installed in the order department situated several miles from the building where the central system is located. A 2701 Data Adapter Unit and modems will be used to connect the 2848 to a system channel via a leased line.

### 5.16.2 LOCAL 2260 SYSTEM

Customer Specifications

There are 6,000 customer records to be stored, of 200 bytes each, and 10,000 items, of 300 bytes each. Three thousand orders, with an average of five items each, are to be handled daily. The orders can be assumed to be evenly distributed, so that there are no peak hours to be considered. 150 telephone rush orders, with one item each, 100 inquiries per day, and rush orders that come in from the branch offices and customers have to be handled by one or two operators who answer the telephone used for inquiries.

Input/Output Message Formats

The following message formats are shown in Figure 173:

| Message Type | Input/Output to CPU |
|---|---|
| 1. Inquiry request | Input |
| 2. Inquiry answer | Output |
| 3. Order entry | Input |
| 4. Order checking | Output |
| 5. Error correction | Input |
| 6. Acknowledgement | Output |

Terminal Specifications

Regarding the message formats, it is assumed that a numeric keyboard is sufficient. The application is such that alphameric entries are not required. We can, therefore, say that the input speed of the operators is 3 digits per second. We also learn from the message formats that a display with 12 lines of 40 characters each is sufficient. It is assumed that wrong customer and item numbers,

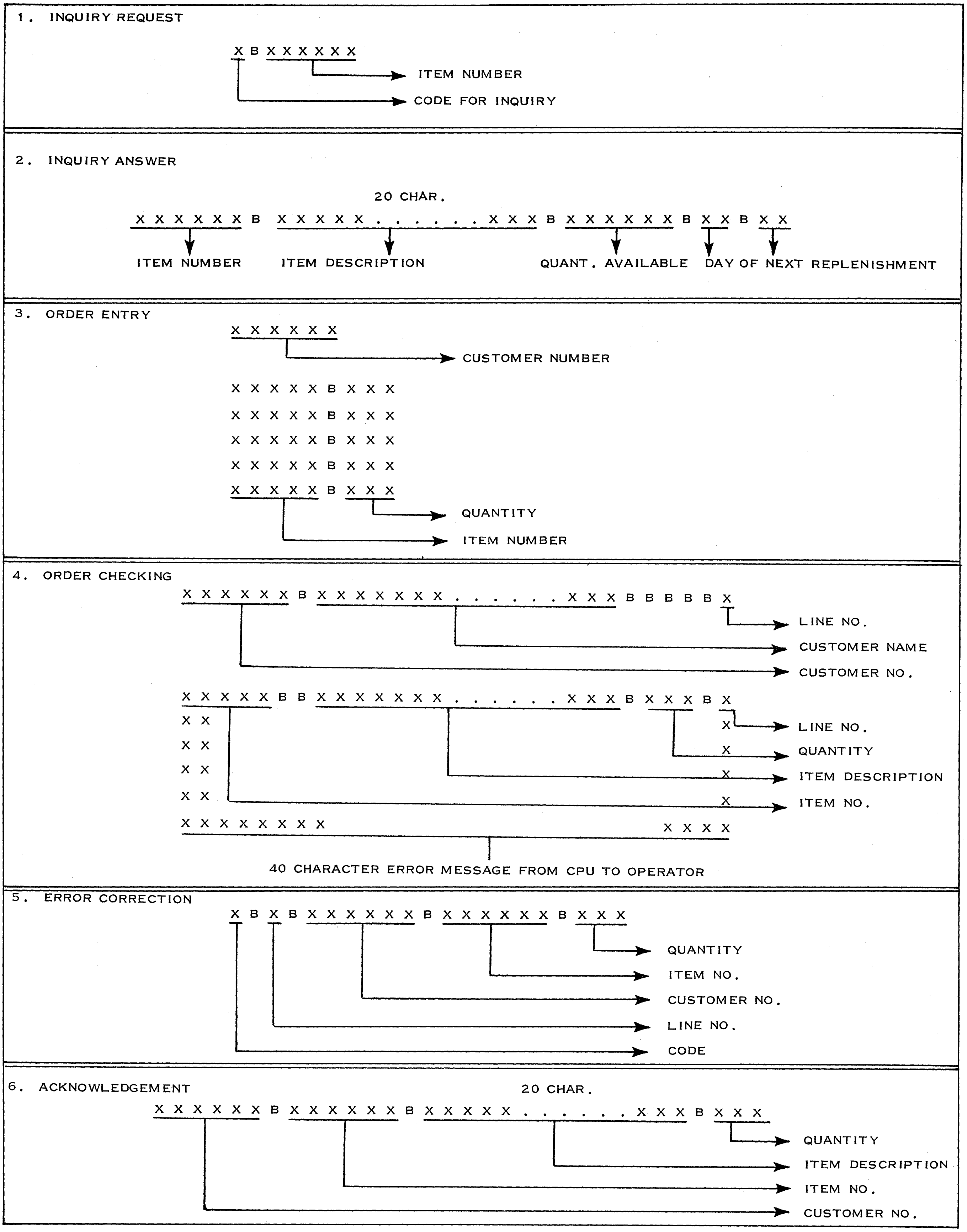


Figure 173. Input/Output Message Formats

as well as keying errors by the operator, will not exceed 1% of the number of lines keyed in. (This may be an unrealistic assumption and is only used here as an example.) This means that in every sixteenth invoice an error occurs that has to be corrected by the operator.

Error Procedure

There are several possible errors that must be corrected by the operator.

1. Errors which the CPU cannot find by checking procedures in the program, i. e. a wrong but available item number has been keyed by the operator. The system will update this wrong item. The error has to be detected by the visual control of the operator. The same is done for the customer number.

2. Errors that are detected by the checking procedures of the CPU, i. e. unavailable customer or item numbers. Here a message is shown to the operator at the second last line on the display. A situation where the quantity ordered is higher than the quantity in stock also falls into this category.

In both cases, the operator has to correct the situation. In the first case, the system has to correct the already updated item before it can process the new item number given by the operator at the time of the last display. It is assumed that the time needed by an operator to check the display for undetected errors, by reading one customer name and five item descriptions is 10 seconds.

The time to handle undetected errors, 1 line of 100, includes looking up books or files, etc., and is an average of 1 minute. This is necessary in 10% of all errors.

Traffic analysis, line load calculation and network design will be discussed in the following subsections.

### 5. 16. 3 REMOTE 2260 SYSTEM

In this example, it has been assumed that the 2260 is installed in a different building several miles away. We can also assume that the 2260's were previously local and that the whole department has been moved. The high data rate of 1,200 characters per second can no longer be used, and we will have to check the effect of this. Transmission is performed between the remote 2848 and the central 2701 on a leased telephone line using 3977 Model 1 modems. Therefore, the speed is 120 characters per second.

All the figures on customer and terminal specifications, message formats, as well as error procedures, will be the same. Traffic analysis and line load calculations have to be studied again.

### 5. 16. 4 TRAFFIC ANALYSIS

Normal Orders

| | |
|---|---|
| Quantity | 3,000 per day |
| Keying speed | 3 characters/ second |

An order is composed of 57 characters (including LF). The total entry of one order takes: $\frac{57}{3}$ = 19 seconds.

Processing Time:

Number of accesses for one order entry: 7
We assume each access takes 150 ms
Disk processing time is: 150 x 7 = 1,050 ms = 1. 05 seconds
We assume 350 ms, including CPU processing time.
Internal response time = 1. 05 + . 35 = 1. 4 seconds.

If we assume a reply time of 4 seconds and checking time of 10 seconds, the total time for one order is:

19 + 1. 4 + 4 + 10 = 34. 4 seconds

As there are 3,000 orders a day, this comes to:
34. 4 x 3,000 = 28 hours 40 minutes for transactions having no errors.

Wrong Orders

Let us compute the additional time taken by recovery of errors. One invoice out of 16 is wrong. On 3,000 invoices, there are $\frac{3,000}{16}$ = 187. 5 erroneous messages.

As the time for one error is 30 seconds, it will take a total of 188 x 30 = 5,640 seconds = 1 hour 34 minutes.

| | |
|---|---|
| Number of important errors | $\frac{188}{10}$ = 19 |
| Time for one error recovery | 1 minute |
| Time for 19 errors | 19 minutes |

Total hold time:
28. 40 + 1. 34 + 0. 19 = 30 hours 33 minutes
The number of working hours is 6 hours per day.
The minimum number of displays would be $\frac{\text{30 hr. 33 min.}}{6}$ = 5. 1

At least 6 displays are then necessary

Rush Orders

Quantity: 150 per day.

The time required is as follows:

| | |
|---|---|
| Key in | 6 seconds |
| Send back a message | 1 second |
| Check | 2 seconds |

Taking into account the time wasted in talking on the phone, the total transaction would take about 1 minute.

The total time is: 150 x 1 = 150 minutes =
2 hours 30 minutes

For telephone inquiries:

| | |
|---|---|
| Key in | 4 seconds |
| Send back a message | 1 minute |
| Checking time | 2 seconds |

The time is about 1 minute, too, because some time is lost in answering the phone, so:

100 x 1 = 100 min. = 1 hour 40 minutes

is necessary for that kind of transaction.

The total time taken for the terminals which work in conjunction with the telephone is:

2 hours 30 minutes + 1 hour 40 minutes = 4 hours 10 minutes.

Two displays are necessary to handle the transactions and have to be reserved for that purpose.

Assuming 6 hours of work per day, the necessary number of display stations is 6 x 2260 + 2 x 2260. Telephone sets are near both the latter.

Terminal Load Calculation

For the 6 normal terminals:

As we have 30 hours 33 minutes of traffic a day (6 hours) for 6 terminals, the load for each will be: $\frac{\text{30 hours 33 minutes}}{6}$ = 305.5 minutes for each.

The load in percent is: $\frac{305.5 \times 100}{360}$ = 85%. According to Figure 158, 6 x 2260's can handle this load. For the terminals with telephones, the traffic time is 4 hours 10 minutes = 250 minutes for 6 hours.

This comes to: $\frac{250}{6 \times 2}$ = 20.8 minutes/terminal/hour. The load in percent for each of these two terminals is $\frac{20.8 \times 100}{60}$ = 35%.

## 5.16.5 LOAD CALCULATION FOR THE CPU

The purpose of this example is to show another type of application for the 2260. Consequently, we will not restate all the computations already made for the 1050 and the remote 2260 stations, but will merely refer to the previous sections.

If we consider the local 2260 application, we can easily see that message types 1 and 2 (inquiry) are very similar to the 1050 inquiry message types or 2260 remote inquiry.

Order entry and checking (message types 3 and 4) are similar to the rush order of the 1050 example.

Error correction and acknowledgement (messages 5 and 6) can also be found in a similar form in the remote 2260 example.

Taking into account the above factors, the total load for the CPU during a peak hour can be reasonably assumed to be 2%. This is based on the fact that each of the message types in this example use the same or similar message processing routines as the corresponding message types in the 1050 and remote 2260 examples. Channel and disk utilization for this are the same as the 1050 example, since similar programs are being used.

SECTION 6.

STR SYSTEM DESIGN

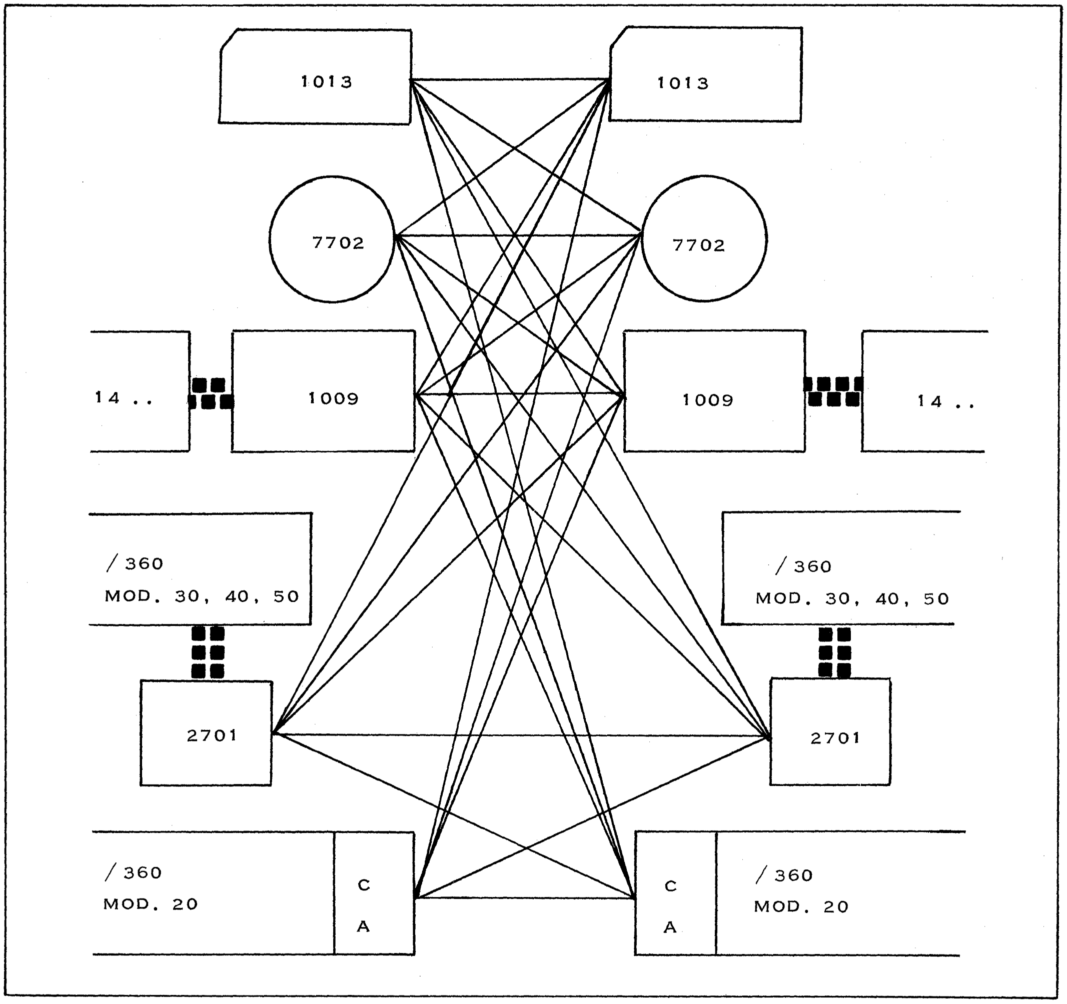


Figure 174. Possible Connections of STR Devices

The STR devices are a group of synchronous transmitter and receiver terminals with different input and output media using the same transmission technique.

This section will discuss the functional characteristics of STR devices, programming support, timings and core requirements, and conclude with an example.

## 6. 1 FUNCTIONAL CHARACTERISTICS OF STR DEVICES

### 6. 1. 1 POSSIBLE CONNECTIONS

Since all STR devices use the same transmission techniques, this provides full compatibility between all devices of the STR family (if operating at the same transmission speed). Figure 174 shows all possible connections of the STR devices. It does not show the complete configuration and all the adapters or features necessary to attach the units to each other, but defines only the attachment capability of various STR devices:

- Connection of STR devices to the communication line;
- Line adapters are not provided to attach STR devices to a communication line;
- Only modems may be used to achieve that operation.

The compatible modems are:

| | |
|---|---|
| 600-1200 bauds | IBM 3977 Model 1 |
| 1200-2400 bauds | IBM 3977 Model 2 |

Other modems may also be used, depending on the regulations of the country concerned (homologation):

| Speed (bps) | Modems | |
|---|---|---|
| 600 | 202C - 202D | leased lines HDX |
| | 202C | switched lines |
| | 202B | leased lines FDX |
| | W. U. 1601A | |
| 1200 | 202C - 202D | |
| | W. U. 2121A | |
| 2000 | 201A | |
| 2400 | 201B | |

### 6. 1. 2 GENERAL FUNCTIONAL CHARACTERISTICS

#### Transmission Speed

600 to 4800 bps, depending on the quality of the lines and the capabilities of the input or output terminals. The most commonly used speeds are: 600, 1200, 2000 and 2400 bps. 4800 bps are used with modems available only on an RSDP basis. (40,800 bits/second high-speed transmission is possible with the SDA on the 2701 and on an RSDP basis with the CA on the 2020.)

#### Communication Code

The STR translates the code from the input device to 4/8 transmission code, which is, after reception, translated into the code of the output device. Thus, two machines using entirely different I/O codes can communicate with each other since both have the common 4/8 language over the communication lines.

#### Parallel/Serial Conversion (Refer to Figure 175)

Data is delivered from the input device serially by character: after translation, the bits are serialized and sent on the line. The receiving device receives and assembles the bits, checks and translates them to the code of the output device, and delivers them parallel-by-bit and serial-by-character to it.

#### Transmission Checking

Two checks are provided in order to detect transmission errors. A character check is made first by the 4/8 code. (A 4/8 character is so designated because there are four 0 and four 1 bits in an 8-bit character -- see subsection 2. 3. 7 of the Handbook.) Another check is made by sending an LRC character (Longitudinal Redundancy Check). This character is formed by both the receiving and the sending terminal by the modulo 2 additions of all 1 bits in each of the 8-bit positions for all data characters of a block. The receiver will compare the received LRC character with its own accumulation.

Other checks provided are: record odd/even checking, I/O checking, and reply conditions, which will be studied in the Error Recovery Procedures subsection (6. 1. 4).

#### Synchronization

In order to recognize the proper bit positions and to form a character, the receiving terminal has to

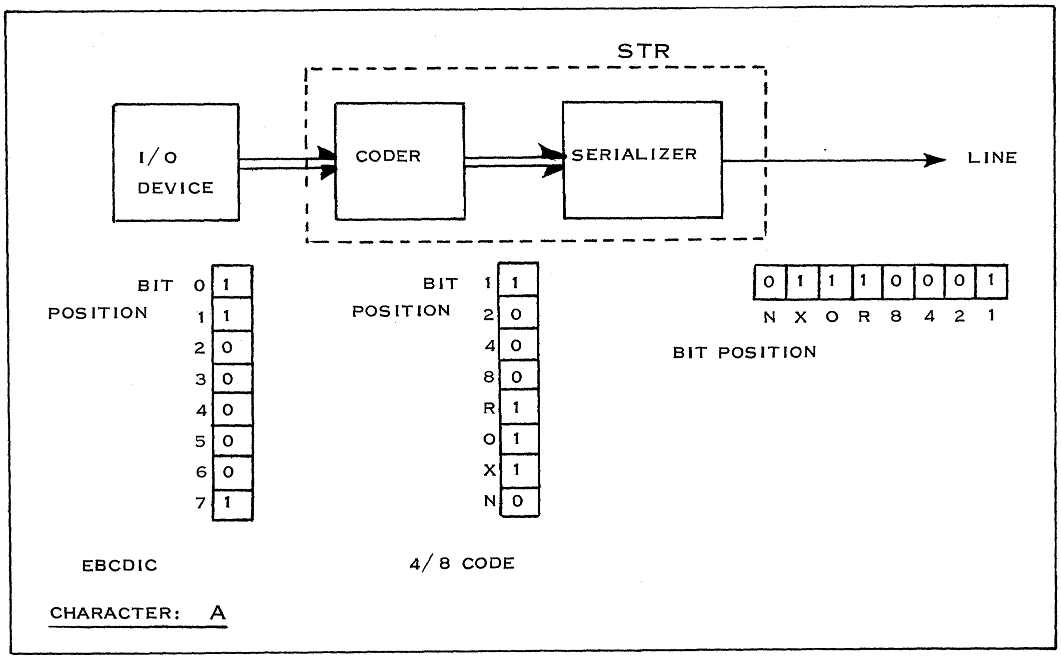


Figure 175. Transmission of the Character "A" to a Communications Line

have a clock (or oscillator) that runs synchronously with the one in the transmitter. Every incoming 1 bit (4 in each character) is used to adjust the receiver clock. This method of maintaining synchronism between the two terminals can work only as long as characters are transmitted on the line. So, if there are no data or control characters to be transmitted automatically, a special character, called IDLE, is sent on the line to make it possible to synchronize the terminals.

Initial character synchronization is achieved in the following way: IDLE characters are sent either in a steady stream (full-duplex operation) or in an alternate manner (half-duplex operation). As soon as each of these characters is recognized, both machines are in character phase; and if the other conditions, such as input/output ready, are completed, the machines will be capable of data communication. For half-duplex operation, the procedure is the same, the difference being that each terminal alternates between sending idle characters and listening.

The transmitting terminal sends IDLE characters for 1. 5 seconds while the receiving terminal listens. The transmitter then sends a turnaround character and becomes a receiving terminal.

Data and Control Characters

In the 4/8 code there are 70 characters, of which 64 are assigned to data and 6 to the control language of the machine.

These six basic characters are as follows:

1. IDLE
2. TL (Transmit Leader)
3. CL (Control Leader)
4. ERR/INQ (Error Inquiry)
5. $ACK_1/SOR_1$ } (Acknowledge/Start of Record)
6. $ACK_2/SOR_2$ }

To allow a wider range of control, the six characters are used generally in two-character sequences. Each sequence is made up of a leader character and a trailer character.

A summary of the control sequences and the functions they perform follows:

IDLE is sent on the line if no data characters are available to maintain synchronization.

CL-IDLE (End of Control) is used by one terminal to transfer control to the other terminal (turnaround).

TL-INQ (Inquiry) is used by a terminal when it wishes to transmit a message.

TL-SOR (Start of Record) is transmitted

immediately before each block of data. The start of Odd Records ($SOR_1$) is transmitted before the first, third, fifth, etc., record of each message, while the start of Even Record ($SOR_2$) is transmitted before the second, fourth, etc., record.

TL-LRC (End of Record) is sent immediately after each record and contains the LRC character which is used to check for transmission errors.

CL-ACK (Acknowledgement) is sent by the receiving terminal after correct receipt of a block of data. The ACK and SOR must be compatible from the parity standpoint. This assures the sender that the receiver has not lost a record.

Acknowledgement is also used as a "Yes" reply to an inquiry.

CL-ERR (Received Error) is sent by a receiving terminal if it receives a block of data which has an error. This sequence notifies the transmitter that it should repeat the transmission of the last record.

CL-EOT (End of Message) is sent by the transmitting terminal after it has transmitted the last record.

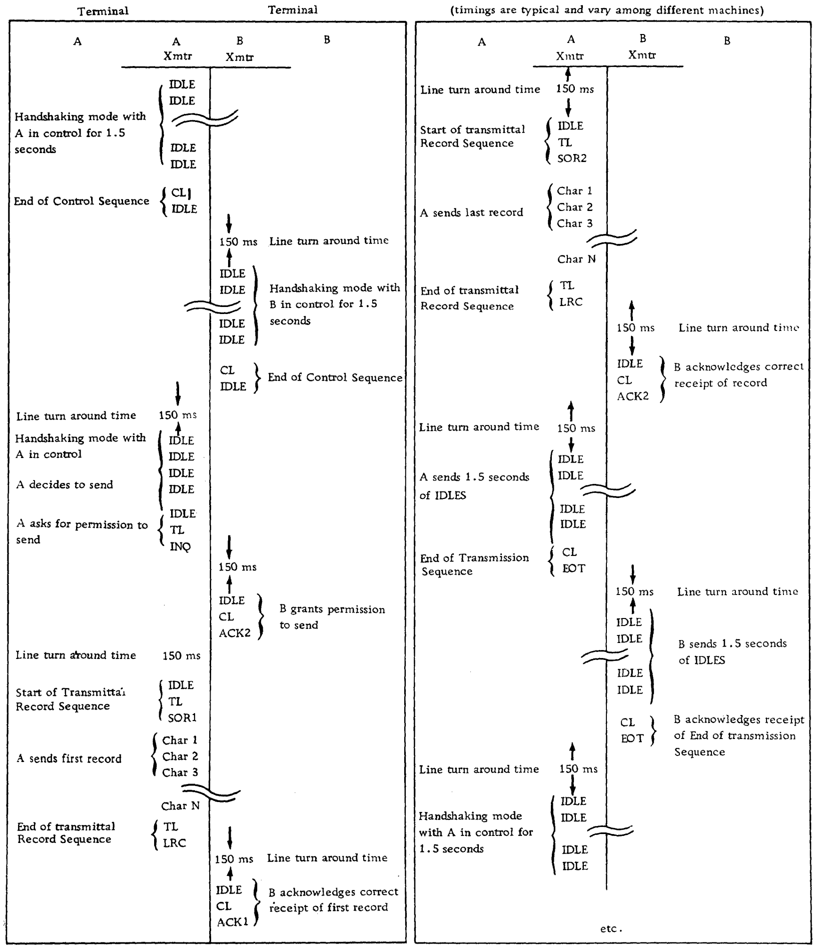
CL-TEL (Telephone) indicates that the terminal operator desires voice communication via the telephone handset with the other terminal operator.

Note: EOT and TEL are special data characters (Y and ] ).

### 6. 1. 3 LINE PROCEDURE AND OPERATION

Figures 176 and 177 show the detailed line procedure for half-duplex and full-duplex operation.

For a complete hardware description, the reader is referred to CEMI: IBM STR Models 1 and 2, Form Z23-6953.



Terminal
Terminal
(timings are typical and vary among different machines)
A
A Xmtr
B Xmtr
B
IDLE
IDLE
Handshaking mode with A in control for 1.5 seconds
IDLE
IDLE
End of Control Sequence
CL
IDLE
150 ms
Line turn around time
IDLE
IDLE
Handshaking mode with B in control for 1.5 seconds
IDLE
IDLE
CL
IDLE
End of Control Sequence
Line turn around time
150 ms
Handshaking mode with A in control
IDLE
IDLE
A decides to send
IDLE
IDLE
A asks for permission to send
IDLE
TL
INQ
150 ms
IDLE
CL
ACK2
B grants permission to send
Line turn around time
150 ms
Start of Transmittal Record Sequence
IDLE
TL
SOR1
A sends first record
Char 1
Char 2
Char 3
Char N
End of transmittal Record Sequence
TL
LRC
150 ms
Line turn around time
IDLE
CL
ACK1
B acknowledges correct receipt of first record
Line turn around time
150 ms
Start of transmittal Record Sequence
IDLE
TL
SOR2
A sends last record
Char 1
Char 2
Char 3
Char N
End of transmittal Record Sequence
TL
LRC
150 ms
Line turn around time
IDLE
CL
ACK2
B acknowledges correct receipt of record
Line turn around time
150 ms
A sends 1.5 seconds of IDLES
IDLE
IDLE
IDLE
IDLE
End of Transmission Sequence
CL
EOT
150 ms
Line turn around time
IDLE
IDLE
B sends 1.5 seconds of IDLES
IDLE
IDLE
CL
EOT
B acknowledges receipt of End of transmission Sequence
Line turn around time
150 ms
Handshaking mode with A in control for 1.5 seconds
IDLE
IDLE
IDLE
IDLE
etc.

Figure 176. Half-Duplex Operation

### 6. 1. 4 ERROR RECOVERY PROCEDURES

The error recovery procedure for STR free-standing units is completely automatic and is made by the hardware. For the adapters (either Model 20 Communications Adapter or the 2701 adapters), the procedure is part hardware and part programming.

In the following section, the error procedure is studied only from the line and terminal standpoints.

Error Detection

A. 4/8 validity check, previously explained in Transmission Checking, subsection 6. 1. 2.

B. LRC check, previously explained as above.

C. Record checking:

If the receiver fails to detect an SOR and the record is less than 3 seconds in length, it will maintain character phase, but will not reply at the end of the record. The transmitter will send IDLES and terminate with the INQ sequence in order to get a reply. The receiver replies with the last reply (i. e. the reply for the previous record). If this reply is other than the RECEIVED ERROR sequence, it will not match the odd or even count of the SOR. This constitutes an unsatisfactory reply and the transmitter will retransmit the record as if it were a RECEIVED ERROR sequence.

If a lost or duplicate record condition occurs, the SOR and ACK cannot be compatible, although the transmitter attempts to retransmit the record. After n attempts, the machine stops and alerts the operator. This location must be regarded as a possible lost or duplicate record location.

D. I/O Checking (Where Applicable):

Errors detected by the input device at the transmitting terminal cause the terminal to terminate the record with the End-of-Idle sequence. Errors detected by the output device of the receiving terminal, when the reply conditions are fulfilled, cause the receiver to reply with the RECEIVED ERROR sequence.

E. Time Out:

If the transmitter, for one reason or another, does not get a reply after three seconds, it will inquire into the status of the receiver by sending TL-INQ. If the reply is satisfactory, the transmitter will go on to the next record; if it is not satisfactory (i. e. the wrong acknowledgement or an erroneous reply), it will retransmit the last record.

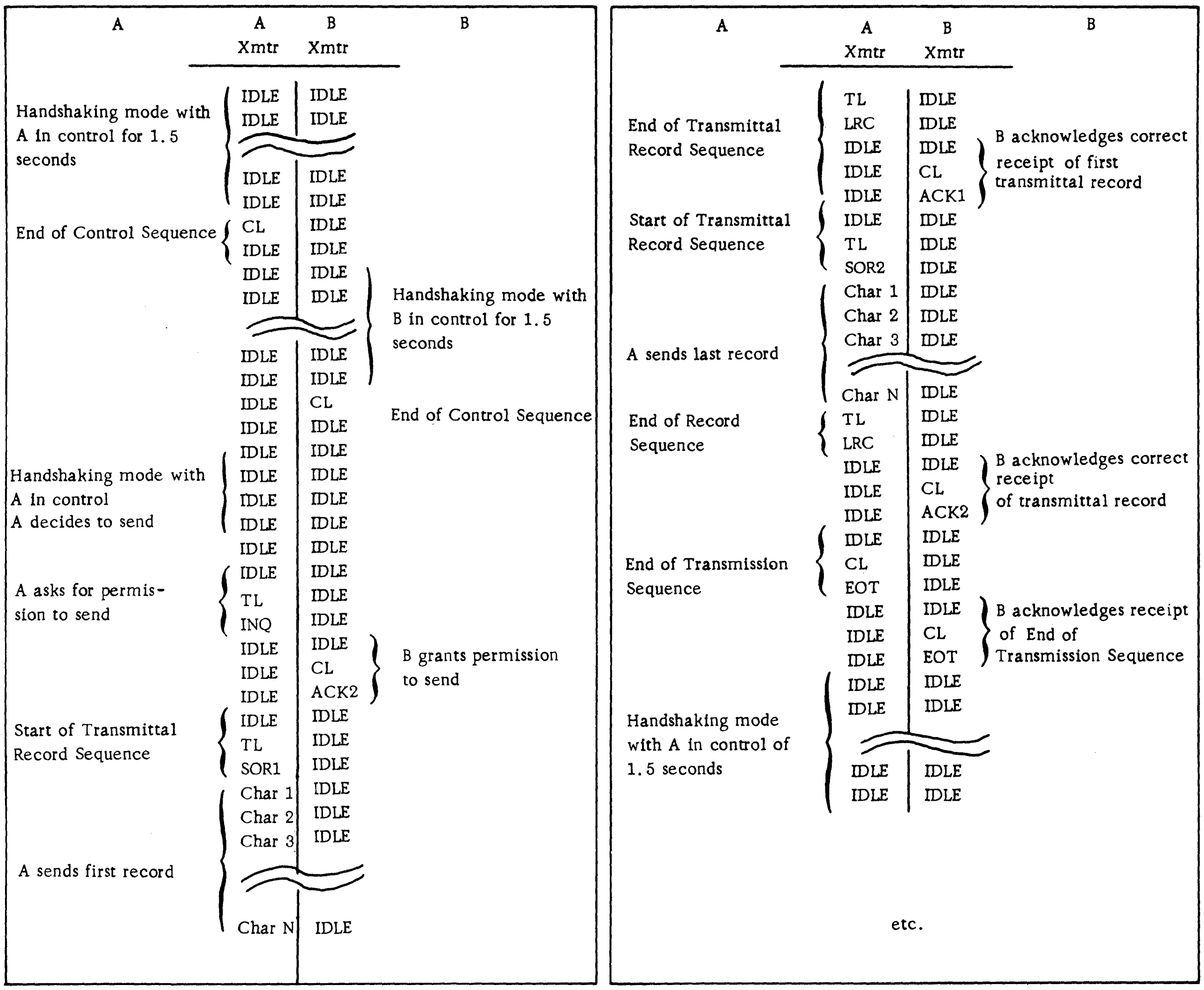
F. Accidental Disconnect:

This does not result in loss of data. When the transmission path is reestablished, character and bit phase will automatically be reestablished. The machines will continue to operate from the last point of contact.

Error Correction

All the errors mentioned above cause retransmission of the entire transmittal record.

(timings are typical and vary among different machines)



| A | A Xmtr | B Xmtr | B |
|---|---|---|---|
| Handshaking mode with A in control for 1.5 seconds | IDLE | IDLE | |
| | IDLE | IDLE | |
| | IDLE | IDLE | |
| | IDLE | IDLE | |
| End of Control Sequence | CL | IDLE | |
| | IDLE | IDLE | |
| | IDLE | IDLE | Handshaking mode with B in control for 1.5 seconds |
| | IDLE | IDLE | |
| | IDLE | IDLE | |
| | IDLE | IDLE | |
| | IDLE | CL | End of Control Sequence |
| | IDLE | IDLE | |
| Handshaking mode with A in control A decides to send | IDLE | IDLE | |
| | IDLE | IDLE | |
| | IDLE | IDLE | |
| | IDLE | IDLE | |
| | IDLE | IDLE | |
| A asks for permission to send | IDLE | IDLE | |
| | TL | IDLE | |
| | INQ | IDLE | |
| | IDLE | IDLE | B grants permission to send |
| | IDLE | CL | |
| | IDLE | ACK2 | |
| Start of Transmittal Record Sequence | IDLE | IDLE | |
| | TL | IDLE | |
| | SOR1 | IDLE | |
| A sends first record | Char 1 | IDLE | |
| | Char 2 | IDLE | |
| | Char 3 | IDLE | |
| | Char N | IDLE | |

| A | A Xmtr | B Xmtr | B |
|---|---|---|---|
| End of Transmittal Record Sequence | TL | IDLE | |
| | LRC | IDLE | |
| | IDLE | IDLE | B acknowledges correct receipt of first transmittal record |
| | IDLE | CL | |
| | IDLE | ACK1 | |
| Start of Transmittal Record Sequence | IDLE | IDLE | |
| | TL | IDLE | |
| | SOR2 | IDLE | |
| A sends last record | Char 1 | IDLE | |
| | Char 2 | IDLE | |
| | Char 3 | IDLE | |
| | Char N | IDLE | |
| End of Record Sequence | TL | IDLE | |
| | LRC | IDLE | |
| | IDLE | IDLE | B acknowledges correct receipt of transmittal record |
| | IDLE | CL | |
| | IDLE | ACK2 | |
| End of Transmission Sequence | IDLE | IDLE | |
| | CL | IDLE | |
| | EOT | IDLE | |
| | IDLE | IDLE | B acknowledges receipt of End of Transmission Sequence |
| | IDLE | CL | |
| | IDLE | EOT | |
| Handshaking mode with A in control of 1.5 seconds | IDLE | IDLE | |
| | IDLE | IDLE | |
| | IDLE | IDLE | |
| | IDLE | IDLE | |

etc.

Figure 177. Medium Speed FDX

## 6. 2 TYPICAL CONNECTIONS AND METHODS OF USE

### 6. 2. 1 TYPICAL CONNECTIONS

It should be mentioned first that leased-line connection between two offline STR devices or dial-up connection between a central offline STR device and some remote offline STR devices is possible, but is not covered here in detail. We will discuss mainly such online STR devices as the 2020 with communications adapter (CA) or the System/360 with 2701 and synchronous data adapter (SDA). Three kinds of system are possible:

1. use as a central processor
2. use as a remote processor terminal
3. use of one of two computers connected by a data channel.

The possible connections are:

a. - leased lines (point-to-point connection)

b. - dial up (functional multipoint sequentially working).

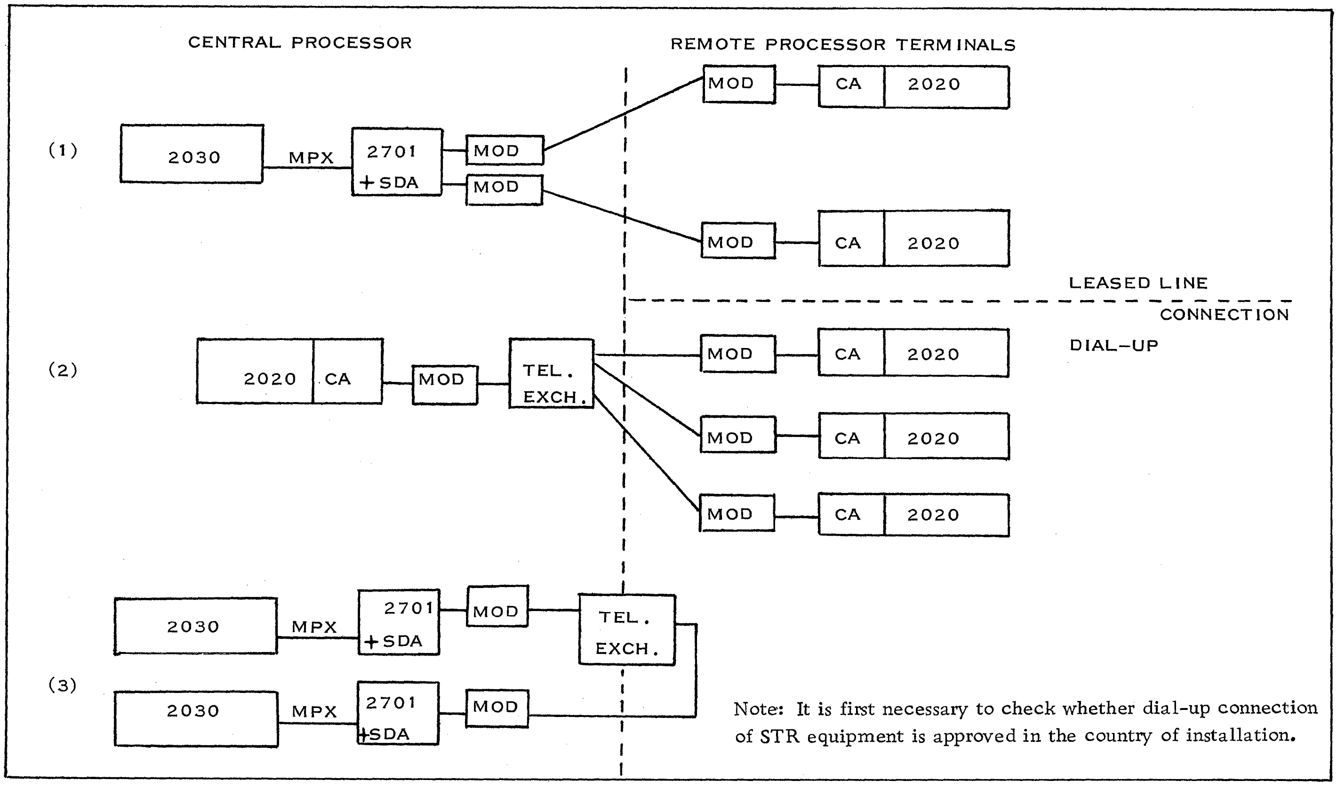
As the central processor in a network of STR devices, only the Model 30 and higher models with a 2701 and SDA can be used when simultaneous operation is desired (see Figure 178). As the Model 20 with CA can connect only one STR line, it must use dial-up connections to different remote STR devices when used as a central processor. In this case, only sequential operation is possible. As a remote terminal, each STR device, offline as well as online (terminal processor) can be used either on a lease-line or a dial-up line basis. Connection of two computers can use leased line or dial-up connection, depending on the amount of data to be transmitted.

Dial-up connections are technically and from the programming viewpoint exactly like fixed connections, once the connection is established. Therefore, the following considerations are limited to fixed point-to-point connections.

### 6. 2. 2 CONFIGURATION (Refer to 3. 6. 4)

A configurator should be used to design a 2701 data adapter unit.

All terminal adapters of the 2701 are divided



Note: It is first necessary to check whether dial-up connection of STR equipment is approved in the country of installation.

Figure 178. Example Showing Connections

into various categories. The STR adapter is called Synchronous Data Adapter Type I (#7696) and belongs in Category II. A maximum of two Synchronous Data Adapters Type I can be housed in the 2701. The second adapter needs the Expanded Capability Feature (#3815) and the Expansion Feature (#3855).

Each Synchronous Data Adapter Type I is connected to one 3977 Modem, Model 1 or 2, or equivalent. There are no IBM line adapters.

Dual Communication Interface Feature (#3462) provides a second data set interface for the Synchronous Data Adapter Type I. The second data set interface is connected by a second 3977 Modem, Model 1 or 2, or equivalent, to a second communication line. Switching between the two interfaces is under program control. This means that a maximum of four STR lines can be attached to one 2701, of which two can operate at the same time.

If the 3977 Model 1 or 2, or equivalent, is used and does not have clocking capabilities, the Internal Clock Feature (#4703) must be specified for each Synchronous Data Adapter Type 1.

### Model 20

To be able to correspond with STR devices, the Model 20 requires the Communications Adapter (#2073), which can be connected only to a 2020 Model B 02, C 02, or D 02.

If both ends of a computer-to-computer connection are to be able to initiate transmission, different techniques have to be used for the Model 20, Model 30, and up. The communications adapter has only two modes of operation: sending and receiving. Thus, first of all, the normal direction of transmission has to be chosen. The sending station is called the master station. In order to enable the other station to initiate a transmission, too, the master station has to send a one-character message which asks the other station whether it has a message to send or not. The answer will be a no (N), or the data itself, starting with the Start-of-Record (TL-SOR) character. This operation is logically the same as polling. Figure 179 describes this operation in detail.

## 6. 2. 3 METHODS OF USE

The STR devices are designed for batch data transmission. The online devices, moreover, are capable of a conversational kind of transmission and can be used for inquiry types of applications.

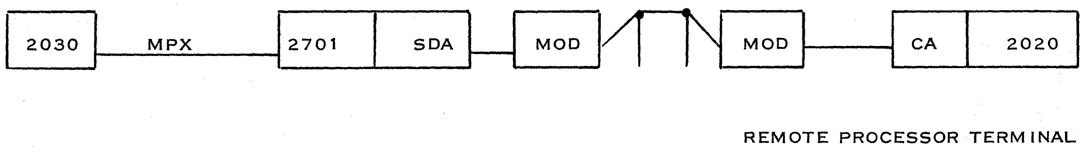
The initiating of transmission is different in both cases and is dependent on the application.

Offline devices can transmit data only in batch, due to the fact that the operator has to initiate every transmission. Online devices have the possibility of initiating transmission by program. If transmission is initiated only by one side (this can include an answering transmission from the other side), there is no problem, as the normal mode of operation is to send on the transmitting and receive on the receiving side.

A transmission can be started at any time by the sending side.

In the case of two Model 30's with 2701 and SDA, this process is much easier, as we have an "uncontrolled mode" in which commands of the attached computer as well as control characters from the remote STR device are accepted. So, in this case, initiation of transmission is possible by both ends without using programmed polling.

Note: These techniques apply only to inquiry-type applications and can be replaced by operator action for a few batch transmissions per day.



| REMARKS | MODE | CHARACTER | MODE | |
|---|---|---|---|---|
| PROGRAMMED POLLING | SEND | ? → | REC | |
| | REC | ← N | SEND | NO DATA TO BE SENT |
| TIME DELAY BY INTERVAL TIMER | | | | |
| | SEND | ? → | REC | |
| | REC | ← SOR | SEND | DATA TO BE SENT |
| | REC | ← DATA | SEND | |
| | REC | ← EOT | SEND | |
| DATA TO BE SENT | | | | |
| | SEND | SOR → | REC | |
| | SEND | DATA → | REC | |
| | SEND | EOT → | REC | |
| | REC | ← N | SEND | |
| TIME DELAY | | | | |
| | SEND | ? → | REC | |
| | REC | ← N | SEND | |
| AND SO ON . . . . | | | | |

Figure 179. Programmed Polling

## 6. 3 PROGRAMMING SUPPORT

As the STR devices are not supported by BTAM, a special access method has been written for them. This STR Access Method (STRAM) is a Type II program. STRAM is used only on a system which has a 2701 with SDA connected. A model 20 with CA must be programmed with the Communications IOCS.

### 6. 3. 1 STRAM

The synchronous transmit-receive access method provides a macro instruction level of support for the transmission and reception of data using the synchronous data adapter (SDA).

Using the macro instructions and routines provided, the user can write programs for transmission between two CPU's as well as for communication with other STR devices via the IBM 2701 SDA, on either a leased line or dial-up basis.

Macro instructions and routines are provided for environment definition, line control, data transmission and code translation. Auto dialing and auto answering are supported. Error recovery routines control restart after detected line errors.

Using the STRAM macros and routines, the user can write programs to establish line connection by dialing or answering, or prepare a leased line for transmission. Data transmission is accomplished through the use of macros and routines which create and execute the necessary channel programs. To convert the 4/8 code for processing, the user issues a TRANSLATE macro. Likewise, TRANSLATE macros are provided to convert System/360 code to 4/8 code for transmission.

STRAM does not provide any form of disk queuing or segment processing. The data is presented to the user for processing without examination (other than for code validity).

#### Implementation with DOS II

The decision to use STRAM must be made at the time the DOS is generated. PUB's (Physical Unit Blocks) and space in the LUB table for assignment of SYS symbols to the devices must be reserved during the system-generation process for the 2701 SDA. Also, the specification TP = BTAM should be included to provide the channel-end appendage capability required. After the SYSGEN process, the STRAM macro definitions must be entered in the user's macro library and the STRAM routines must be placed in the relocatable library.

#### Implementation with OS/360 (SPS)

During generation of the operating system, UCB's must be generated for the 2701 SDA's. After the SYSGEN process, the STRAM routines must be added to the user's LINK LIB, and the STRAM macro definitions must be entered into the user's macro library. In addition, the multiple wait feature of the OS/360 is required.

#### Supported Terminals (or Computers)

A System/360 Model 30 or up programmed with the STR Access Method can communicate with the following devices:

- System/360 Model 30 and up with 2701 and SDA
- System/360 Model 20 with CA
- 1978 (normal, column binary, send or receive first character)
- 1009 Data Transmission Unit (connection to 1400 and 7000 Series machines through 1414's
- 7701/2 Magnetic Tape Terminal
- 7711 Data Communication - Magnetic Tape
- 1974 Terminal
- 1013 Card Transmission Terminal

STRAM macro instructions are classified as follows (refer to Figure 180):

a. environment definition

b. line control

c. basic read/write

d. chained read/write (only in OS version)

e. translation

Note: Besides the basic language, which deals with transmission of single records with specified data areas, there is a second level in the OS version of STRAM which utilizes command and data chaining to provide a continuous flow of data on the line without I/O interruption. This chained level provides buffering capability as well as record segmentation.

|  | OS | DOS | ACTION |
|---|---|---|---|
| A) | SDCB<br>OPEN<br>CLOSE | DTFSTR<br>DO NOT<br>EXIST | DEFINE DATA CONTROL BLOCK. DEFINE THE FILE.<br>PREPARE THE SDCB FOR PROCESSING.<br>DISCONNECT STR DATA SET FROM USER'S PROGRAM. |
| B) | SINCH<br>SCODE<br>SOFF | SAME | INITIALIZE ADAPTER FOR OPERATION.<br>CREATE A CODED PARAMETER LIST FOR DIALING.<br>TURN OFF AN ADAPTER. |
| C) | SREAD<br>SWRITE<br>SEOT<br>STEL<br>SWAIT | SAME | READ A TRANSMITTAL RECORD.<br>WRITE A TRANSMITTAL RECORD.<br>SEND AN EOT SEQUENCE TO THE REMOTE TERMINAL.<br>SEND A TEL SEQUENCE TO THE REMOTE TERMINAL.<br>WAIT FOR AND TEST COMPLETION OF STR OPERATIONS. |
| D) | SINIT<br>SGETR<br>SGETS<br>SPUT<br>SPUT EOT<br>SGETBUF<br>SFREEBUF<br>SEOR | DO NOT<br>EXIST | INITIALIZE LINE CONTROL BLOCK FOR CHAINED OPERATION.<br>STR GET TRANSMITTAL RECORD.<br>STR GET SEGMENTED TRANSMITTAL RECORD.<br>STR PUT A TRANSMITTAL RECORD<br>STR TRANSMIT EOT SEQUENCE AFTER ALL BUFFERS ARE FREE.<br>STR OBTAINS A BUFFER.<br>STR RETURN BUFFER TO POOL.<br>STR SET END OF RECORD. |
| E) | TTOBCD<br>TFMBCD<br>TTO 78<br>TFM 78<br>TTO 360<br>TFM 360<br>TTO FCM<br>TFM FCM | SAME | TRANSLATION TO FROM<br>BCD 1 : 1 CHARACTER<br>COLUMN BINARY FOR 1978 1 : 2 CHARACTER<br>EBCDIC 1 : 2 BLOCKS<br>FIRST CHAR. MODE FOR 1978 1 : 1 CHARACTER |

Figure 180. STRAM Macros and Their Functions

```
STR       SDCB        DDNAME = DD1, EXLST = EXIT, MACRF = RING,
                      BUFNO 5, BUFL = 400

          SOPEN       STR

          SINCH       STR, 1, TYPE = OUT, SPEED = X, DUPLEX = 2 HD
                      TRANS = TTOBCD

          SWAIT       1 . 1

          SINIT       STR, 1, RING = 5, TYPE = OUT, SEG = NO

NEXT      SGETBUF     STR, 1

          MOVE DATA INTO BUFFER

          TTOBCD      STR, XAREA, 400

          SPUT        STR, 1, 400

          B           NEXT

END       SEOT        STR, 1, 9

          SWAIT       1 . 1

          SOFF        STR, 1

          SCLOSE      STR
```

Figure 181. Examples of Chained Transmission to a Line

### Chained Operation

Continuous transmission of data on the line is obtained by constructing a channel program which uses data and command chaining to avoid channel interruption. The channel program is looped into a "ring channel program", which can continue to execute without interruption as long as the user processes the buffers quickly enough.

The ring channel program is constructed by the SINIT (Initialize STR) macro instruction. SINIT initializes the ring channel program and, in the case of input transmission, obtains a ring full of buffers. Buffering can be achieved in one or two ways:

1. At SDCB time the user can supply buffer parameters -- BUFNO, BUFL, BUFAIN -- which will be used at OPEN time to construct a buffer pool for the STR DCB. The buffers from this pool can be used by any line in the STR program.

2. Should it be beneficial for a particular line not to use the buffers of the DCB buffer pool, the line may have specified for itself at SINIT time its own buffer parameters and a buffer list (list of addresses of the buffers supplied by the user. ) SINIT will then chain these buffers together and they will be used as an individual pool for this line.

It is possible for every line to provide its own buffer list. However, in order to conserve core and provide exchange buffering, the DCB buffer pool should be used whenever possible.

Input records are brought in, in advance of program requests, and are placed in buffers. The channel can, however, queue only a ring full of buffers. If the channel fills all the buffers specified in the ring channel program, channel operation will stop for that line until the user takes a buffer for processing, at which time the channel operation is again started.

Output records are presented to the ring channel program by the user and transmitted in order of presentation. Each CCW in the ring channel program can hold one buffer. Thus, the access method will "queue" up to a ring full of buffers. If the ring channel program cannot handle a buffer, the buffer will be refused.

### Example of Chained Transmission to a Line

Figure 181 is an example of chained transmission to a line.

The main part of this program is the loop starting with NEXT. With SGETBUF we reserve a buffer for line 1 (parameter 2) specified in the DECB named STR (parameter 1). After having moved the data into this buffer, it is translated into the 4/8 code used for transmission.

We assume here that only the BCD subset of the EBCDIC character is used. This allows one-to-one character translation. With the SPUT macro, the CCW related to this buffer is chained to the ring channel program for transmission. If the maximum ring size is reached, the program will wait until a buffer is transmitted. Instead of giving control to a background program by this wait, a branch to service another line is possible. If filling of the buffers is faster than the transmission (which is the normal case), continuous transmission is achieved. If processing of data and filling it into the buffers takes more time than the transmission, chained operation does not have any advantage over single-record transmission on a read/write level.

### Message Transmission in Segments

The criteria for using SGETR to receive whole records or SGETS to receive record segments are:

1. If records are fixed in length and buffers can be provided to handle the records, then SGETR should be used.

2. If records are variable in length, but the variance from the smallest possible record to the largest is relatively small, and if available buffers are large enough to hold even the largest records, then SGETR should be used as well.

3. In any other situation, SGETS should be used.

To inform STRAM of the length of the current record, a SEOR macro instruction must be issued after the first segment of each variable record has been received. This allows STRAM to move into the channel program as quickly as possible and turn on the CC (Command Chain) bit in place of the CD (Chain Data) bit, thus preventing the normal interrupt at the end of the record, and keeping the flow of data on the line continuous. At the time the SEOR is issued, the channel program has already progressed into the filling of the next buffer connected with the next CCW. Thus, it is too late to change the flag bits in that CCW; and the earliest that STRAM can effectively change a flag bit is in the third CCW connected with each record. So, in order to maintain a continuous flow of data, each variable-length record must use a minimum of three buffers.

All this does not mean that records smaller than three buffers in length cannot be received. It simply means that there will be a channel end, necessitating a restart of the channel program, thus reducing overall line efficiency.

Code Translation

Since the SDA does not provide code translation by hardware, this has to be done by the program. The 4/8 code used for transmission has 70 possibilities. So only the 64 BCD character subset of the System/ 360 EBCDIC code can be translated one-to-one. If the 256 EBCDIC character set is used, every System/ 360 character must be coded by two 4/8 characters. This is done blockwise (per transmittal record). For the binary mode of the 1978, the two 4/8 characters representing a binary character (column) have to follow each other. As this translation requires more time, a special TRANSLATE macro is provided. If the 1978 works in first character mode, depending on the first character, translation to or from BCD or binary is performed by the TTOFCM or TFMFCM macros.

## 6. 3. 2 CIOCS

The Communications Adapter Input/Output Control System provides the user with a set of tested routines to control the transmitting and receiving of data by the Model 20 with Communications Adapter. All standard STR terminals and adapters can be used as remote stations.

Similar to the IOCS for punched card equipment, the CIOCS consists of a generator and a set of routines which the user tailors to his requirements by means of definition statements. The programmer uses macro instructions in his source program to perform the desired input/output operations and some control functions. These macro instructions result in macro expansions in the assembled program, which link to the appropriate CIOCS routines. These, in turn, are assembled either separately or jointly with the user's program after being generated by the CIOCS generator, based upon the definition statements provided by the user. IOCS and CIOCS routines must always be generated together, with the IOCS preceding the CIOCS routines.

Definition Statements

Definition statements are used by the programmer to define:

- the set of data to be transmitted or received;
- the terminal or adapter communicating with the Model 20;

| MACRO INSTRUCTION | FUNCTION |
|---|---|
| OPEN | SETS INDEXES, SETS COMMUNICATIONS ADAPTER TO TRANSMIT OR RECEIVE, ETC. |
| GET | MOVES NEXT LOGICAL RECORD FROM INPUT (RECEIVE) AREA INTO USER'S WORK AREA, TRANSLATES FROM 4-OF-8 INTO MACHINE CODE, INITIATES RECEIVING IF INPUT AREA IS EMPTY, ETC. |
| PUT | MOVES A LOGICAL RECORD FROM USER'S WORK AREA INTO OUTPUT (TRANSMIT) AREA, TRANSLATES FROM MACHINE TO 4-OF-8 CODE, INITIATES TRANSMITTING IF OUTPUT AREA IS FULL (NOT NECESSARILY IN THIS ORDER). |
| CLOSE | ENSURES PROPER HANDLING OF THE DATA TO BE TRANSMITTED AFTER ALL RECORDS HAVE BEEN PROCESSED. |
| TRUNC | TRANSMITS A "TRUNCATED" BLOCK , I. E., AN INCOMPLETELY FILLED OUTPUT AREA. IT ALSO TRANSMITS COMPLETELY FILLED OUTPUT AREAS IF A HEADER HAS BEEN SPECIFIED. |
| RELSE | SKIPS THE MOVING OF RECORDS REMAINING IN THE INPUT AREA AND INITIATES RECEIVING OF THE NEXT BLOCK. |
| CHEAD | CHANGES THE CONTENTS OF ROUTING OR ANSWERING CODE FIELDS IN STANDARD BLOCK HEADERS. |
| BRAHE | BRANCHES TO A USER-SPECIFIED ADDRESS WHENEVER A STANDARD BLOCK HEADER CONTAINS A CERTAIN ANSWERING OR ROUTING CODE, OR AN EOT INDICATION. |

Figure 182. CIOCS Macro Instructions and Their Functions

- the mode of operation (send or receive);
- the required I/O areas;
- other details.

The generator reads these statements and, based upon the information contained in them, selects the routines required for the user's particular application.

Macro Instructions

These symbolic instructions are used in the main source program to perform the desired input/output operations and some control functions. They result after assembly in macro expansions that link to the previously defined CIOCS routines.

Macro instructions enable the user to design his program free from most detailed considerations concerning transmitting and receiving. Moreover, the programmer need be concerned only with processing of individual logical records. The filling or emptying of his input/output areas, blocking or unblocking, insertion or deletion of padding characters and overlapping are handled in accordance with the information given in the definition statements by the CIOCS routines.

Figure 182 is a listing of all available macro instructions and their functions.

Sample Coding for a Simple Transmit Program with CIOCS

Problem Definition:

Punched cards have to be transmitted in batch from a Model 20 to an IBM 7702 Magnetic Tape Transmission Terminal. All 80 columns of a card are to be transmitted. 12 cards should be transmitted in one block of 960 characters. Two output areas should be provided to overlap processing and transmission. Figure 183 shows the coding for this example.

```
NAME       OPERATION   OPERAND                           COMMENTS            72

TRCA       DTTSR
                       DEVICE   =  TTTO2,                                    C
                       TYPTMN   =  TRMT,                                     C
                       BLKHDR   =  NONE,                                     C
                       WORKA    =  YES,                  DEFINITION          C
                                                         STATEMENTS
                       10 AREA1 =  OUT 1,                                    C
                       RECFORM  =  FIXBLK,                                   C
                       RECSIZE  =  80                                        C
                       BLKSIZE  =  960
           DTTEN
TMT PRO    START       1024
                                                         INITIALIZATION OF
           OPEN        READ                              TMT PROGRAM
           OPEN        TRCA
TR 1       GET         READ, WORK
                                                         MAIN LOOP (ANY
           PUT         TRCA, WORK                        PROCESSING HAS TO
                                                         BE INSERTED HERE)
           BC          15, TR 1
EOF        CLOSE       READ
           CLOSE       TRCA                              PROGRAM END
```

Note: Refer to System/360 IOCS for the Communications Adapter, C26-3606; and System/360 Model 20 IOCS for the Communications Adapter, Operating Procedures, C24-9004.

Figure 183. Sample CIOCS Coding

## 6. 4 EFFECTIVE LINE SPEED AND THROUGHPUT

Effective line speed is the number of data characters transmitted over a line per second. It is dependent upon:

- block length
- kind of line and modems used
- error rate of the line

BLOCK LENGTH

As the number of control characters per block is a constant, the effective speed increases with the length of the block. The number of control characters per block is 8: IDLE, TL, $SOR_1$ (data), TL, LRC (turnaround time), IDLE, CL, $ACK_1$ (turnaround time).

KIND OF LINE

On half-duplex lines, two turnaround times have to be inserted (one before and one after the acknowledgement). This time is dependent on the modems used:

IBM 3977 turnaround time is: 70 ms = 10. 5 (150 characters/second) character times.

The formula for calculating the effective line speed for an error rate of 0, therefore, is:

T = turnaround time in seconds
S = theoretical speed in characters/second
N = number of data characters per block
E = effective speed

$$E = \frac{NS}{N + 8 + 2ST}$$

ERROR RATE

The error rate is normally expressed in A errors per $10^x$ characters or $A \times 10^{-x}$. These errors occur mostly in bursts or are at least randomly distributed. For the following figures, in the worst case a fixed inter-arrival time for errors is assumed to ease the calculations. For an error rate of $10^{-5}$ and a block length of 1,000 characters, for example, every hundredth block has to be retransmitted. This effect is shown in Figures 184 and 185. The formula becomes:

$$E = \frac{NS}{N + 8 + 2ST} \times (1 - A \times 10^{-x} \times N)$$

Explanation of the Graphs

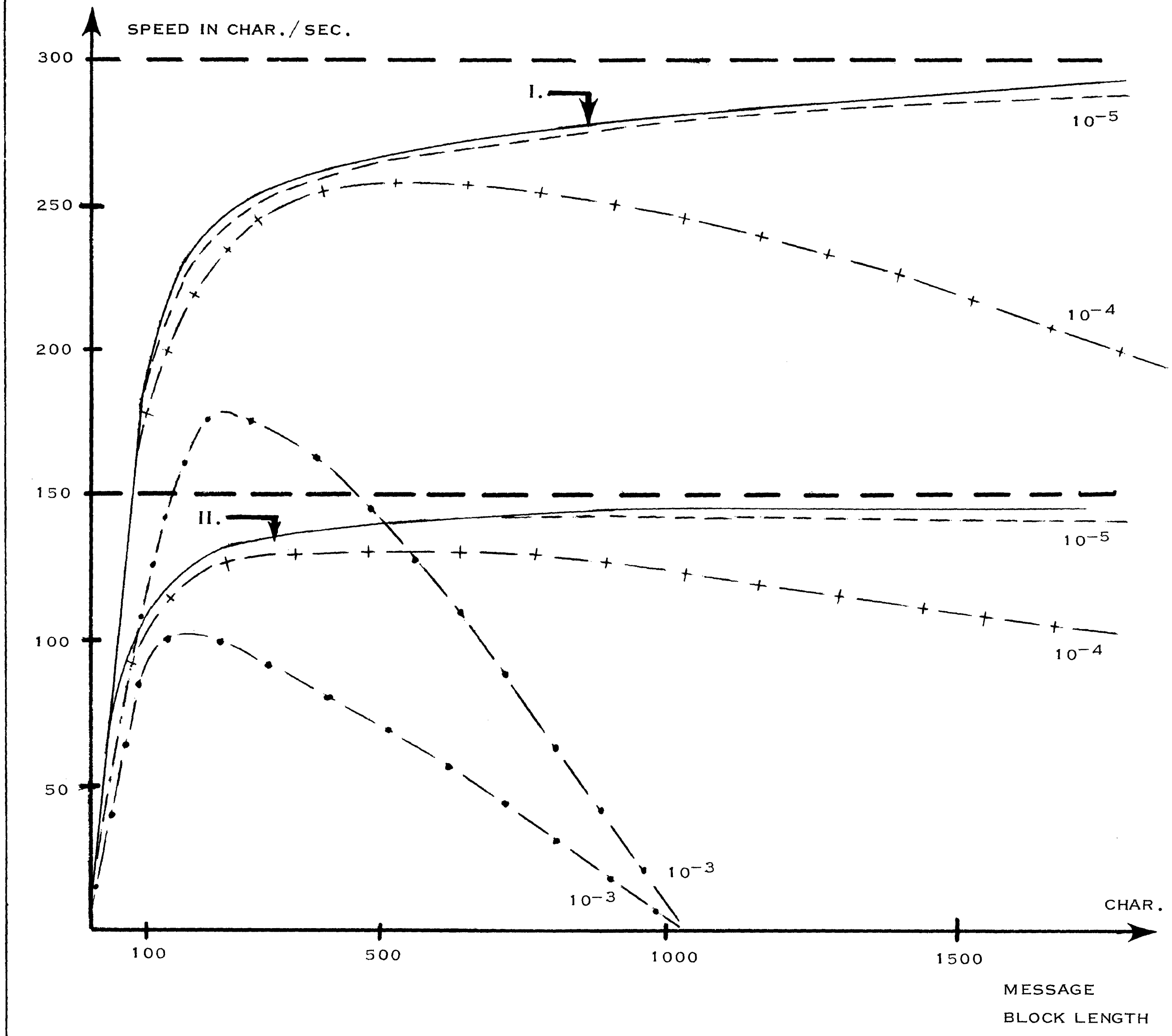
The first graph (Figure 184) shows the effective speed in characters/second as a function of different message lengths for half-duplex operation. We have chosen a 3977 Modem with a turnaround time of 70 ms. The curve I ( - ) shows the speed on a perfect line, i. e. free of transmission errors. The ( - + - + ) curve is given for a transmission error rate of $10^{-4}$, and the ( - . - . ) curve is given for a transmission error rate of $10^{-3}$. This curve does not take into account the propagation time due to the line, which is negligible, but does take into account a total turnaround time of 140 ms.

Figure 185 represents the same effective speed for a full-duplex operation. At this time, the modem turnaround time no longer applies, but we assume the adapter turnaround time as one-character time. The delay due to propagation time of the transmission line is assumed as 2 to 5 ms for a 500 km line.

Propagation time is the time taken by an electric signal to go from one end to the other end of a line. This time is dependent on the physical constitution of the line and the length. This time can be roughly estimated as 1 ms per 100 km.

REDUCTION OF THROUGHPUT EFFECTED BY I/O OPERATIONS

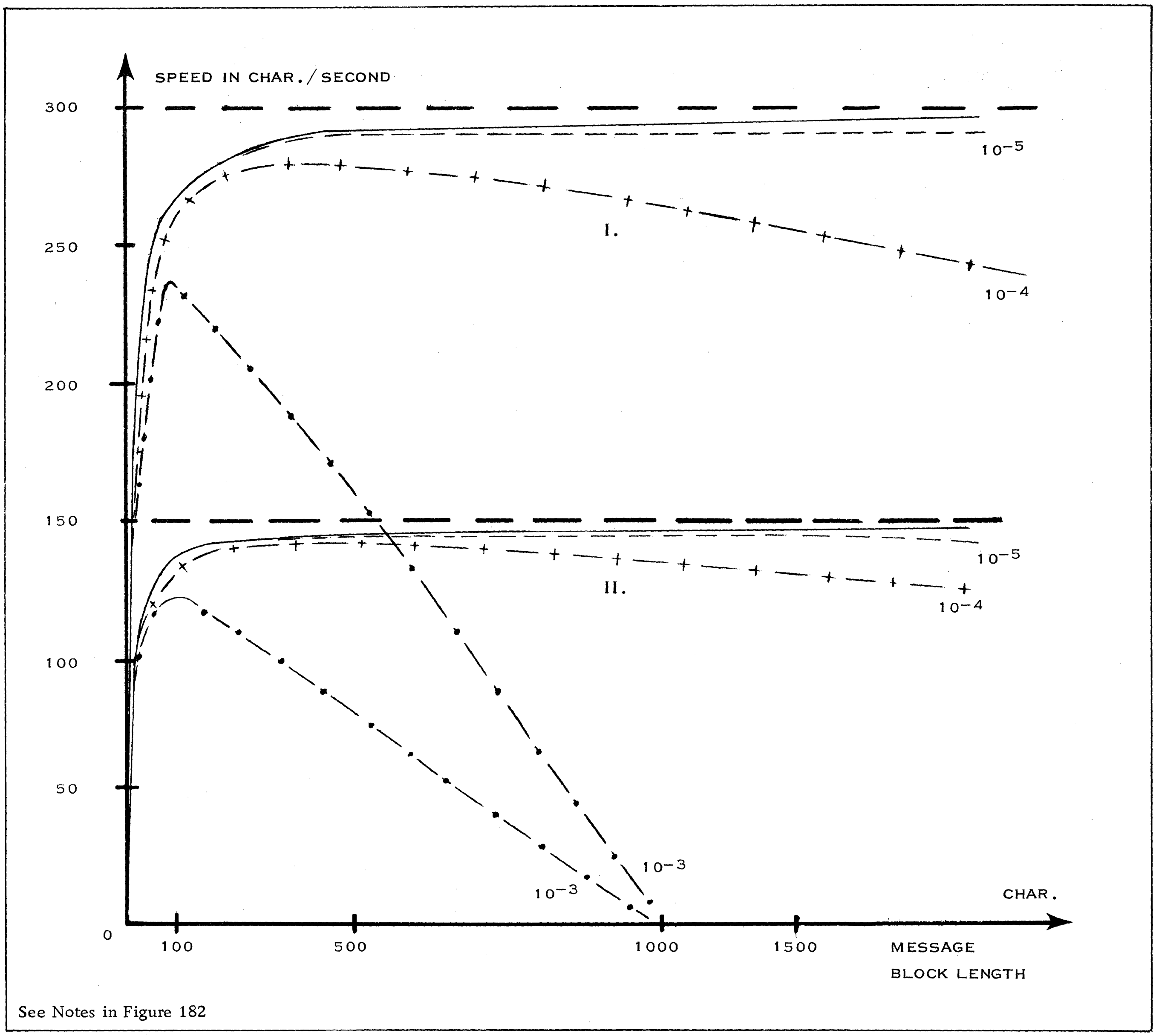
Sometimes a printer or card unit of an STR terminal or terminal processor cannot accept or deliver data from or to the line as fast as it is transmitted. For example, a 10 lines/second printer on a Model 20 with CA which prints only 10 characters of the received data per line (plus perhaps some additional data) can accept an effective speed of only 100 characters/second. This must be taken into account in throughput analysis.



Notes: The broken lines show the effect of the message error rate on the effective speed.
- 3977 Modem (70 ms turnaround time)

Theoretical curve without errors ————
With $10^{-5}$ message error rate — — —
With $10^{-4}$ message error rate ——+——
With $10^{-3}$ message error rate ——•——•——

Figure 184. Effective Line Speed: Half-Duplex Operation



See Notes in Figure 182

Figure 185. Effective Line Speed: Full-Duplex Operation

## 6.5 CPU TIME AND CORE REQUIREMENTS

### 6.5.1 COMPUTER SLOWDOWN BY CHANNEL INTERFERENCE

The figures shown in Figure 186 were obtained from Average Reduction in Processing Capability Caused by Channel Interference, Form Z20-1780. These are maximum figures. They apply to long blocks and full-duplex lines with short turnaround times.

The figures for 150 characters/second transmission speed are about 50% less.

Sample Calculation

Assumptions:

Chained operation (start I/O and channel end times can be disregarded).
Half-duplex line, turnaround time = 150 ms.
Theoretical speed: 300 characters/second.
Block length = 100 characters + 8 control characters.
All other traffic on the channels is negligible.

Calculation:
Time to transmit one block:
108 x 3.33 + 2 x 150 ms = 660 ms.
Interference time for one block:
(Control characters are not transferred to the CPU).
Sel. Ch. Mod. 30:

$$\begin{aligned} 100 \times 1.5\ \mu s &= 150\ \mu s \text{ data transfer} \\ &\underline{\ \ 39.75}\ \mu s \text{ command chaining} \\ &\ \ 190\ \mu s \end{aligned}$$

CPU utilization therefore is $\frac{190\ \mu s}{660\ ms} = 0.000288$

or 0.3%.

MPX Ch. Mod. 40:

$$\begin{aligned} 100 \times 28.8\ \mu s &= 2880\ \mu s \text{ data transfer} \\ &\underline{\ \ \ \ \ 75}\ \mu s \text{ command chaining} \\ &2955\ \mu s \end{aligned}$$

CPU utilization therefore is $\frac{2955\ \mu s}{660\ ms} = 0.00448$

or 0.45%.

| CHANNEL INTERFERENCE | MODEL 30 | MODEL 40 | MODEL 50 |
|---|---|---|---|
| 300 CH/S. ON MPX | 1.6 | 0.8 | 0.4 |
| ON SEL | 0.05 | 0.04 | 0.03 |

Figure 186. Channel Interference

### 6.5.2 TIMING ESTIMATES FOR LINE CONTROL AND MESSAGE PROCESSING

Line Control

Chained Transmission:

As we have only one START I/O and one channel end for the whole transmission, these can be disregarded. The time for the transmission of one block is composed of the execution times of the three macro instructions SGETBUF or SFREEBUF, T ... (translate) and SPUT or SGET and one branch instruction.

For the Model 40, this is:

| | | |
|---|---|---|
| SGETBUF (SFREEBUF) | = | A µs |
| T TO BCD (or other) | = | N x 625 + 16.88 µs |
| SGET (or other) | = | B µs |
| BRANCH | = | 9.38 µs |
| | T + | (N x 625 + 26.26 + A + B) µs |

with N = number of characters per block.

If we add these times which result in T and divide the transmission time of one block, which is $\frac{N}{S}$, with S = speed in characters/second, we get the maximum CPU utilization due to data transmitted over one STR line:

$$*U = \frac{T \times 10^{-6} \times 100 \times S}{N}$$

Note: $10^{-6}$ used to convert T into seconds
100 used to get U in %.

The total CPU utilization during a peak hour is:

$$U_T = \frac{(\text{No. of Transactions/peak hr.}) \times (\text{Time/transaction})\ \text{sec.}}{3600}$$

Single Record Transmission:

If we disregard, as in the example above, initialization and closing of transmission, basically the three macro instructions, SWAIT, SREAD (SWRITE) and T ... (translate), a branch instruction must be considered for timings:

*This formula, if multiplied by the number of STR lines in the configuration, would give the maximum utilization of a system with 100% line loading.

| | |
|---|---|
| SWAIT | A μs |
| SREAD (SWRITE) | B μs |
| TRANSLATE | C μs |
| BRANCH | D μs |

$$T = A + B + C + D$$

The formula for calculating the maximum CPU utilization is, again (in percent) due to data transmitted over one STR line:

$$U = \frac{T \times 16^{6} \times 100 \times S}{N} = \frac{T \times S \times 10^{-4}}{N}$$

Message Processing

Timing estimates for message-processing programs can be based, in most cases, only on the number of instructions and the average instruction time, since the exact programs are not available. This is shown in subsection 3. 12.

### 6. 5. 3 CORE ESTIMATES

Maximum core storage for the STR Access Method is:

6 ± 2K for the DOS version

and

8 ± 2K for the OS version

(Core requirements for the different macros will be included as soon as they are available. )

The number of instructions for the message-processing program has to be estimated and multiplied by 4. 5, the average instruction length.

Refer to subsection 3. 2. 9 for details. See Figure 187 below for examples of core estimates.

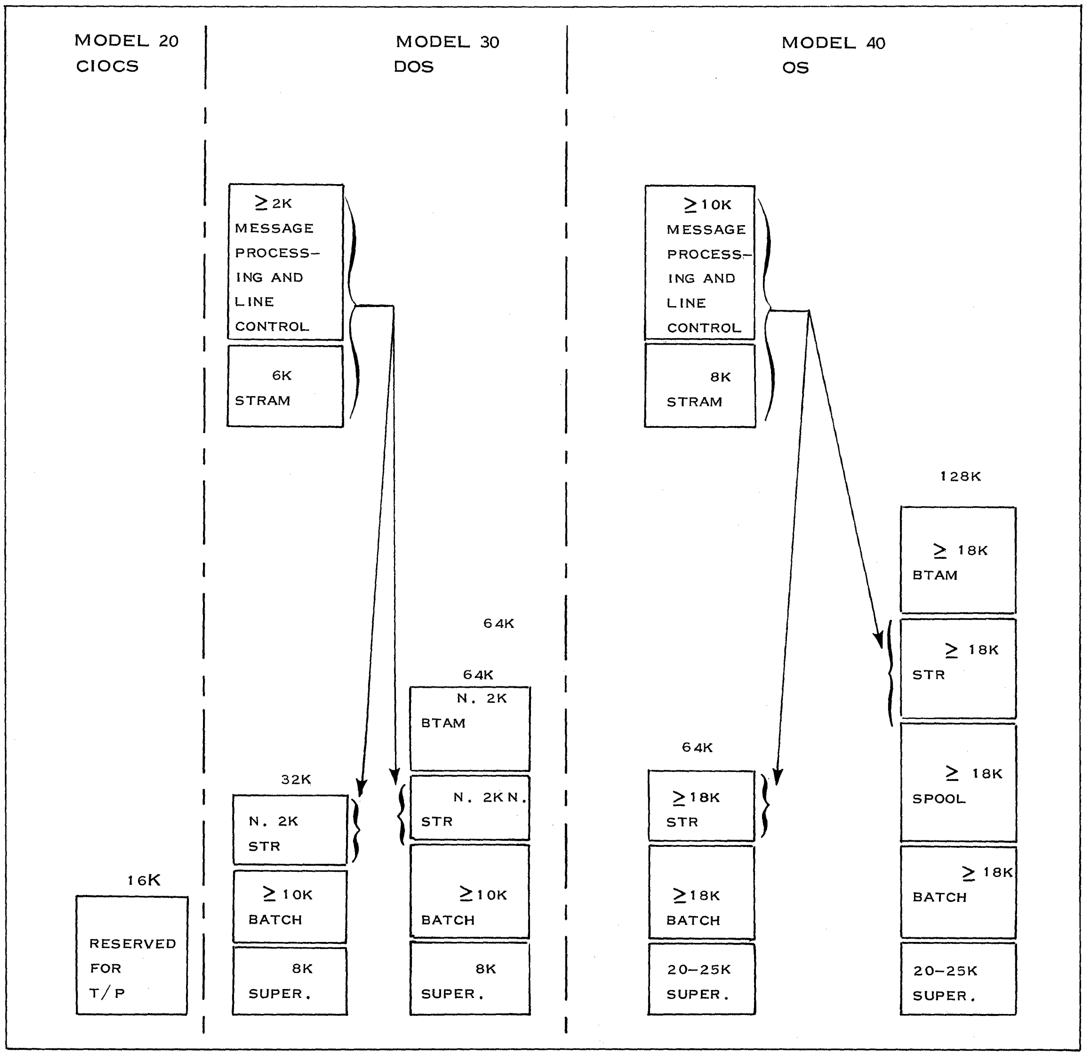
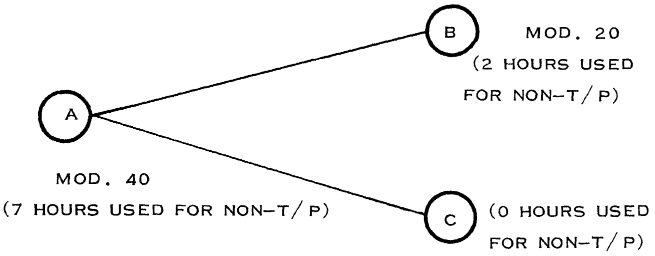


Figure 187. Minimum Core Requirements

## 6. 6 EXAMPLE OF STR SYSTEMS

In order to define a general example, we will not use a special application, but simply give the assumed locations, the installed or planned equipment for non-T/P use at these locations, and the traffic between them:


B
MOD. 20
(2 HOURS USED FOR NON-T/P)
A
MOD. 40
(7 HOURS USED FOR NON-T/P)
C
(0 HOURS USED FOR NON-T/P)

Traffic between locations per day:

| | | | |
|---|---|---|---|
| A to | B | 100,000 characters, disk to printer | 5,000 lines |
| A to | C | No traffic | |
| B to | A | 30,000 characters (500 cards) to disk | |
| C to | A | 96,000 characters (1600 cards) to disk | |

All transactions should be done in batch and should not take more than 30 minutes each.

### 6. 6. 1 HARDWARE SELECTION

For card transmission of 96,000 characters in less than half-an-hour (= 1,800 seconds), we need an effective speed of 53 characters/second, which can easily be handled by an IBM 1013 Card Transmission Unit working with a theoretical speed of 150 characters/second (terminal C).

Other transmissions can also be done in less than half an hour if an STR connection with a theoretical speed of 150 characters/second is used.

In the following section, we discuss a configuration of Models 40 and 20 to establish an STR connection.

2040 Configuration

Since two lines are necessary for this application, the Model 40 needs a 2701 Data Adapter Unit with two Synchronous Data Adapters Type I (#7696). To attached the second SDA, Expanded Capability Feature (SF #3815) and Expansion Feature #3855 are prerequisites.

2020 Configuration (Refer to Figure 188)

Transmission Adapter #2073 provides the connection of the transmission line to the 2020 Model BO 2, CO 2, or DO 2.

### 6. 6. 2 CALCULATION OF TRANSMISSION TIMES

Basic assumptions can be made as follows:

Modem: IBM 3977 Model 1
Speed (theoretical): 1200 bps (150 characters/sec.)
Modem turnaround time: 70 ms
Mode of operation: half-duplex
Line error rate (character): $10^{-4}$

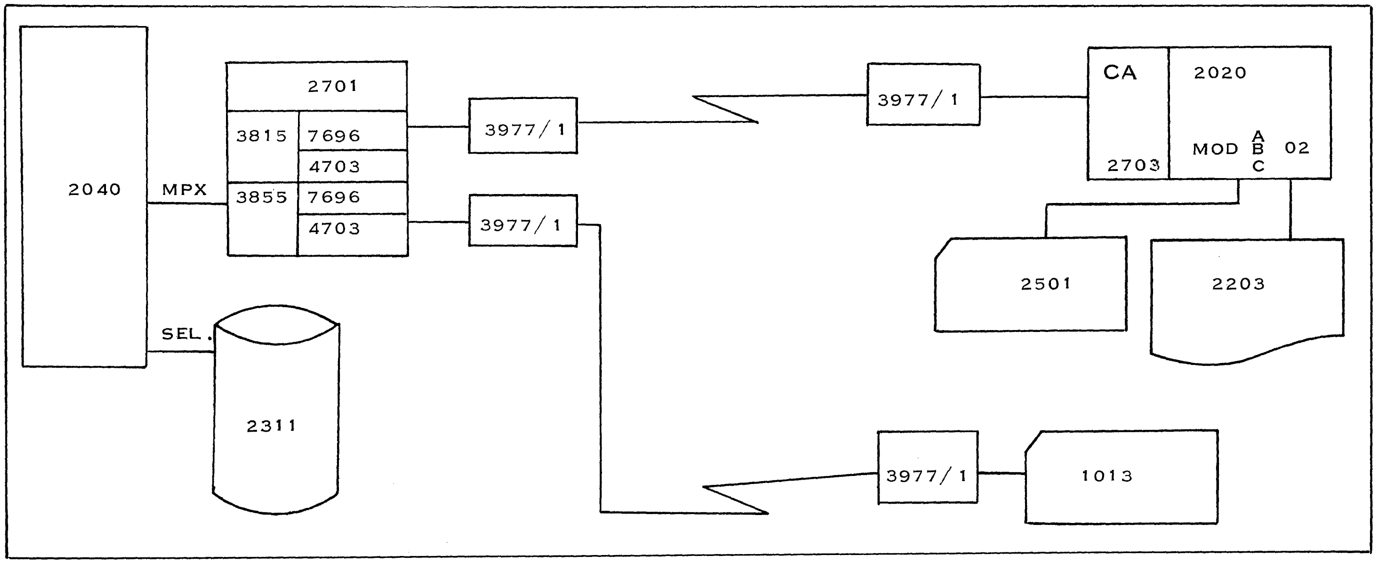


Figure 188. Hardware Configuration: 2040-2020

The printer used is a 2203 Model A 01, which prints at 350 lines/minute with a 52-character set.

Calculation of Effective Line Speed

First, the optimum length of the blocks has to be determined. From the curves given in Figure 184, we can see that the optimum block length for a character error rate of $10^{-4}$ is about 400 characters. The curve gives the corresponding effective line speed, too: 140 characters per second.

Disk-to-Printer Time (Model 20)

The time necessary to transmit $10^5$ characters (or 250 blocks of 400 characters) is: $\frac{10^5}{140} = 715$ seconds, or 12 minutes. But we must check whether this speed is not decreased by the limitations due to the I/O unit: in this case, the printer. We assumed the print speed is 350 lines/minute. Since we have to print 5,000 lines, it will take $\frac{5,000}{350} = 14$ minutes 17 seconds to print those lines. This will be the actual transmission time.

Model 20 Card-to-Disk Time

We assume the card reader is a 2501 Model A 01 with a reading speed of 10 cards per second. Obviously, the line is slower than the I/O device and the line speed can therefore be used to compute the time: $\frac{30,000}{140} = 215$ seconds, or 3.6 minutes. Total transmission time on the line between the Model 20 and the Model 40 is: 12 + 3.6 = 15.6 minutes. The initial requirement (half an hour) is therefore met.

Card-to-Disk (1013)

Again, only the line speed need be considered. It takes: $\frac{96,000}{140} = 685$ seconds ≈ 12 minutes. For this line also, the operation is achieved in less than 30 minutes.

Calculation of CPU Utilization for the Model 40

As calculated in subsection 6. 5. 1, the CPU interference for the Model 40 due to transfer of data and command chaining is negligible and only the CPU interference due to the program has to be taken into consideration. The formula for calculating the maximum CPU utilization for the line control program given in subsection 6. 5. 2 is:

$U = \frac{T \times 10^{-6} \times 100 \times S}{N}$. For the 1013, the maximum record length is 329, and we will use this figure for N. T for the chained operation on the Model 40 is assumed to be 6 ms. $U = \frac{6,000 \times 10^{-4} \times 150}{329}$

$U \approx 0.3\%$ for the line control program

The message-processing program which reads records from disk, formats them, sends them out, and checks for proper transmission, is very simple, using about 2K bytes for instructions. Using the figures of 4.5 bytes/instruction and 35 µs/instruction given in subsection 3. 8, this program needs an estimated time of $\frac{2,000 \times 35}{4.5}$ µs ≈ 15 ms per block of 329 characters, which is a time period of 329 ÷ 150 = 2.2 seconds. This results in utilization of the CPU for message processing of $\frac{15 \text{ ms}}{2,200 \text{ ms}} \approx 0.007 \approx 0.7\%$. Total CPU utilization for transmission of one block of data on one STR line with 1,200 bits/second and a block length of 329 is consequently 0.3 + 0.7 = 1% during the time period of the transmission of the block.

The CPU utilization during the 30 minutes for 565 four-hundred-character messages is:

$$= \frac{565 \times .021}{1800} \text{ sec.}$$

$$= 66 \times 10^{-2}\%$$

$$= .66\%$$

Sample Program for the Model 40

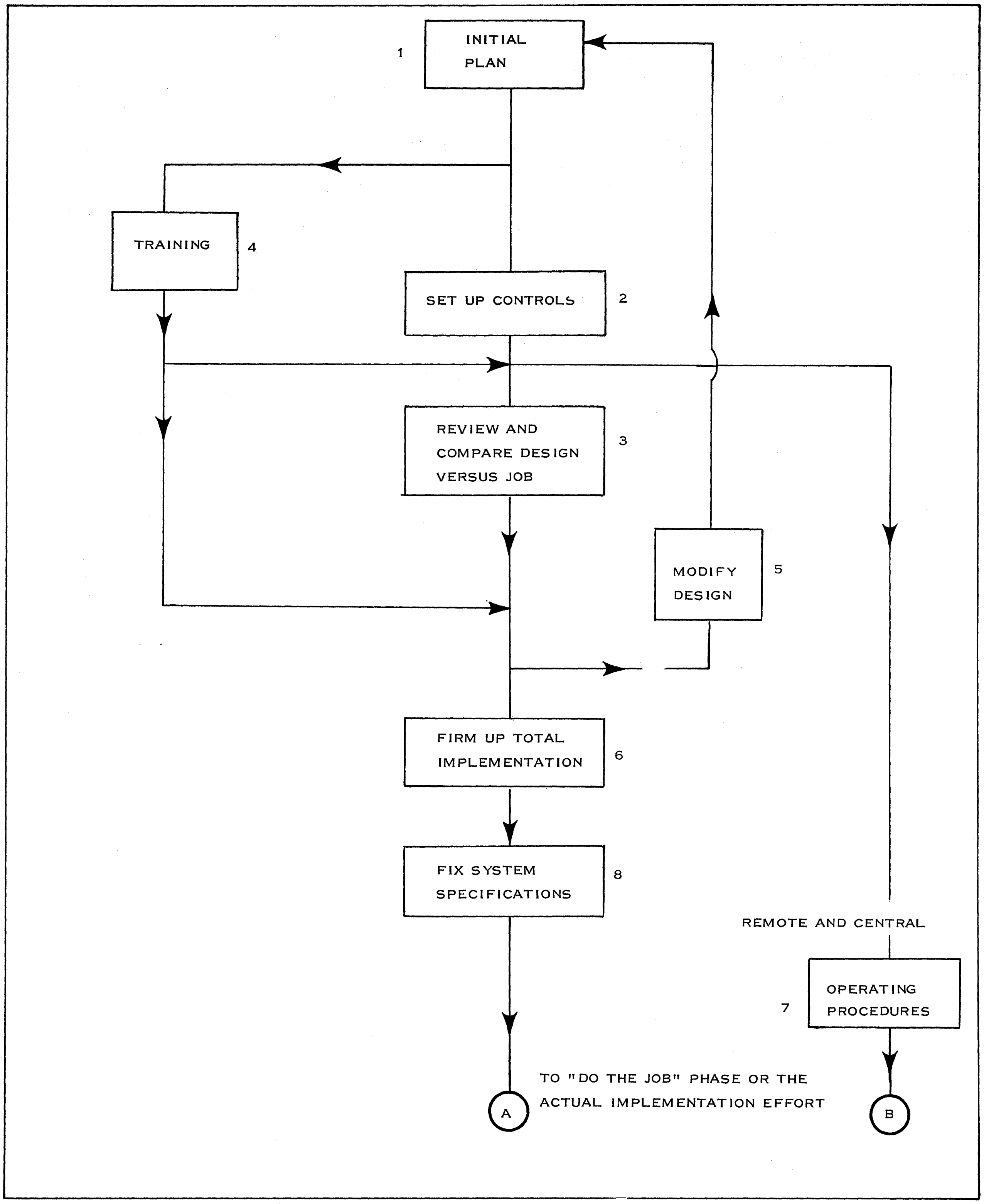
This sample program uses STRAM in OS-SPS to receive the 500 cards (80 columns punched and transmitted) in blocks of 960 characters and puts them on disk. The sample program is shown in Figure 189.

```
RCVPRO     CSECT
           BALR        2, 0
           USING       *, 2
           B           START
STRM 20    SDCB        DDNAME = DD1, EXLST = EXIT
DISKFI     DCB         DSORG = PS, MACRF = (W), DDNAME = DD2
EXIT       DC          X ' 02 '                       CONDITION CODE FOR EOT
           DC          AL3 (EOTAD)                    ADDRESS FOR EOT ROUTINE
           DC          X ' 03 '
           DC          AL3 (TELAD)                    ADDRESS FOR TEL ROUTINE
           DC          X ' 81 '
           DC          AL3 (ERREND)                   ADDRESS FOR ERR ROUTINE
IAREA      DS          CL960
OAREA      DS          CL960
START      OPEN        STRM 20                        OPEN STR LINES
           OPEN        DISKFI                         OPEN DISK FILE
           SINCH       STRM 20, 1, TYPE = IN, SPEED = X,
                       DUPLEX = 2 HD, TRANS = TFMBCD
           SWAIT       LINENO = 1                     WAIT FOR INITIALIZATION
           SREAD       STRM 20, 1, INPUT, 960         RECEIVE DATA
           MVI         ECBC, X ' 7 F '                INITIALIZE DISK ECB
WAIT       SWAIT       LINENO = 1                     WAIT FOR TRANSMISSION
           CHECK       ECBC
           MVC         OAREA, IAREA
           SREAD       STRM 20, 1, INPUT, 960         RECEIVE NEXT DATA
           TFMBCD      STRM 20, OAREA, 960
           WRITE       ECBC, SF, DISKFI, OAREA        PRINT
           B           WAIT
EOT AD     CHECK       ECBC                           WAIT FOR LAST WRITE
ERREND     CLOSE       STRM 20
           CLOSE       DISKFI
           RETURN
TELAD      WTO         ' TEL RECEIVED '
           SREAD       STRM 20, 1, INPUT, 960
           B           WAIT
```

Figure 189. Sample Program for the Model 40

## SECTION 7.

## PRELIMINARY IMPLEMENTATION PLANNING FOR A SMALL CBS



1
INITIAL PLAN
TRAINING
4
SET UP CONTROLS
2
REVIEW AND COMPARE DESIGN VERSUS JOB
3
MODIFY DESIGN
5
FIRM UP TOTAL IMPLEMENTATION
6
FIX SYSTEM SPECIFICATIONS
8
REMOTE AND CENTRAL
OPERATING PROCEDURES
7
TO "DO THE JOB" PHASE OR THE ACTUAL IMPLEMENTATION EFFORT
A
B

Figure 190. Preliminary Implementation Plan

The preliminary implementation plan is an integral part of the system design. It outlines to the customer and IBM what efforts are required to install the system. It enables the customer and IBM to plan ahead in terms of manpower requirements and also supplies them with a reasonable estimate of how much time the installation will require.

## 7.1 INITIAL PLANNING

The first phase of the implementation effort is called initial planning, and the primary tasks that make up this phase are:

1. Review design
2. Develop schedule
3. Assign manpower
4. Develop documentation

The initial planning phase of an implementation is actually a detailed development of the preliminary implementation plan. The relationship of the major tasks within the initial planning phase of the implementation is shown in Figure 190. This phase and the tasks within it must be considered in the preliminary implementation plan.

A summary description of each of the tasks shown in Figure 190 is as follows:

1. Initial Plan

This consists of setting up the implementation team and the implementation plan. Usually, the implementation team is made up of most of the personnel involved in the system design. A poor choice of personnel can materially lengthen the schedule. It is, therefore, most important that the team as a whole should have an in-depth knowledge of the problem and the solution.

2. Set Up Controls

This consists of assigning tasks, developing the schedule, and setting up the documentation plan.

Setting up controls in addition to the schedule includes progress reports, meetings, and checkpoints.

a. Progress Reports:
A system must be set up in which at appropriate intervals the team leader sends progress reports to all concerned. The report should be a brief summary of what was accomplished and what is expected to be accomplished between reports. The reports should also include any information concerning special meetings that have been held. If at all possible, time sheets can be kept so that a good indication of the time each task has taken for an installation effort is available.

b. Approvals:
In a document to all team members, a method for approval of flowcharts and other documents concerned with the programming should be outlined, such as approval by the team leader and IBM leader of flowcharts before coding begins.

c. Meetings:
A schedule, or at least a tentative schedule, for meetings must be set up because time must be allotted to them, and this time does not contribute to the working implementation effort.

d. Checkpoints:
A major contributor to the control of an implementation is the setting up of checkpoints. Checkpoints should be specified for the time when functional specifications are firm, when the general flowcharts should be completed, scheduled completion date of network design, scheduled completion date of evaluated package programs, scheduled completion of coding major routines are established, and at points within the testing and debugging schedule.

e. Documentation:
Documentation is necessary for a successful implementation, especially in the area of program philosophy, flowcharting, and coding. The reasons for documentation are given here:

- Major coordination aid
- Makes possible cross-referencing between team members; defines interface between sections of the system

- Reduces duplication of effort
- Indicates problem areas
- Aids in bringing new people on board
- Speeds up implementation by reducing updating time

The documentation plan is based on documentation specifications and a work book. The specifications contain a description and example of flowcharting and coding. They also show how the overall reference system will work. In addition, they should specify a master file that contains all the major tasks separately. Again, it is necessary to emphasize that documentation is a necessary evil, if you will. Past experience has shown that where CBS installations have been done in a rush and where the people concerned did not take time to document, they always had to reprogram and redesign a second implementation effort after the first one failed.

3. Review and Compare Design vs. Job

As there is a time lag between the acceptance of the proposal and initiating the implementation, a review of the system design is necessary. The implementation schedule must allow for time to review the system design. The following are the reasons for reviewing the design and the items that must be checked during the review:

a. Change in team when proceeding from design to implementation: A review of the design enables the new members of the team to become familiar with the system design and gives them a good basis on which to build their implementation effort.

b. Time lag between proposal and initiating the implementation.

c. Changes in specifications by the customer.

d. Change in traffic.

e. Change in terminals and network.

f. Relocation of the computer center.

g. Announcement of new equipment.

h. Change in RPQ's.

i. Availability of new tariffs, such as multipoint lines or lines operating at high speeds

Time lag between proposal and initiating implementation: This is a major overall reason for review. Because of the time lag, inevitably there are changes that will have an effect on the design. Items c to i are the major ones caused by a time lag.

At this time, variations in optional hardware should be reconsidered.

4. Training

While the controls are being set up and the design is being reviewed, IBM personnel and the customer should be trained and educated.

5. Modified Design

After the review of the design, if the system does not meet the customer's specifications or requirements, a redesign of the system or at least a modification must be made.

6. Firm Up Total Implementation

This is where the implementation team would review the final network design. They would also arrive at the remote and central operating plans, and the remote and central backup plans in detail, based on the operating procedures and backup procedures specified in the system design. In addition, at this point, Special Engineering must be contacted to make sure that the RSDP's are being developed and that schedules can be met.

7. Operational Procedures

Operational procedures for the remote and central site are a continuing effort and must be performed concomitantly with all other tasks.

8. Fix System Specifications

This includes a meeting with Field Engineering and freezing the total implementation plan. This is the most important task of all in the system design. It is the point at which implementation truly begins. In this phase of implementation, the design team must make sure that the specifications are now at such a detailed level that they define all the logical and functional interfaces in the svstem and specify how they will interact to achieve the cost and performance obj ectives of the system. It is mandatory for all systems that these specifications be adequately documented and be in narrative form.

## 7.2 "DO THE JOB" OR THE ACTUAL IMPLEMENTATION EFFORT

The second phase of the implementation is the actual implementation effort. Each of the tasks within this phase must be understood prior to developing a preliminary implementation plan. This phase has as its primary tasks:

1. Evaluation of programming support
2. Coding
3. Testing and debugging
4. Remote terminals
5. Communication facilities
6. Total system test
7. Cutover

Figure 191 shows the actual implementation effort, which we call "do the job".

Almost all the tasks are performed in parallel with the exception of program coding and the debugging plan at the beginning of "doing the job", and the system test and cutover at the end.

A summary description of the tasks shown in Figure 191 is as follows:

### 1. Operator Training

This includes training of both the central and remote operators. Central operator training should be scheduled so that it is completed before the system enters the final system test phase.

One of the most important aspects of any communications-based system implementation is the training of the terminal operators because they are the ones directly responsible for input to the central computer and output from the central. If they fail to do their jobs efficiently, the rental hardware would not be properly justified.

### 2. Network and Terminal Plan and Implementation

This includes the remote site preparation and the installation of the remote hardware. The physical location of the terminals at the various termination points is based on the application and terminal types. Obviously, a banking terminal location does not take as much preparation as a 1030 terminal that will be used in a warehouse environment or if the remote terminal is a small computer.

This task also includes defining the remote physical and logical interface.

The remote physical interface is the confirmation of the physical connections at each remote location. Specific details, such as connector description, cable length, wire and pin assignments, maintenance access to connectors, data set access and power, are necessary to ensure before installation that there will be no problems.

The remote logical interface consists of the implementation team confirming the method of controlling remote devices from the computer. This includes line control functions, such as polling and selecting; features like tabbing and special characters; and actual message formats. The line control system must be documented in great detail, regardless of whether the line is leased or switched. If common carrier (other manufacturers' equipment) telegraph terminals are used, this information will be verified at a meeting with the other manufacturer and the common carrier. The common carrier in most cases, as far as World Trade is concerned, is the PTT.

### 3. Cutover Plan

Cutover occurs when the system is deemed ready for operational use. It is the most crucial stage in the system implementation for both the customer and IBM. It must be done in an orderly manner and therefore requires considerable planning early in the implementation.

### 4. Site Plan and Site Preparation

This is the planning for the central site for the installation of the computer center. Certain things that may be considered in site preparation would be: whether backup power and air conditioning units should be supplied in case of failure; where the common carrier facilities or equipment are located and what types are required; or whether a dehumidifier is necessary.

### 5. Confirm Communications Facilities

This includes defining the interfaces, meeting with the common carriers, and ordering and installing the facilities.

Where the T/P system is in-house, the PTT is not involved. This makes the installing of the remote equipment and communications system simpler than if the PTT were involved. An in-house or in-plant system can use IBM modems and/or line

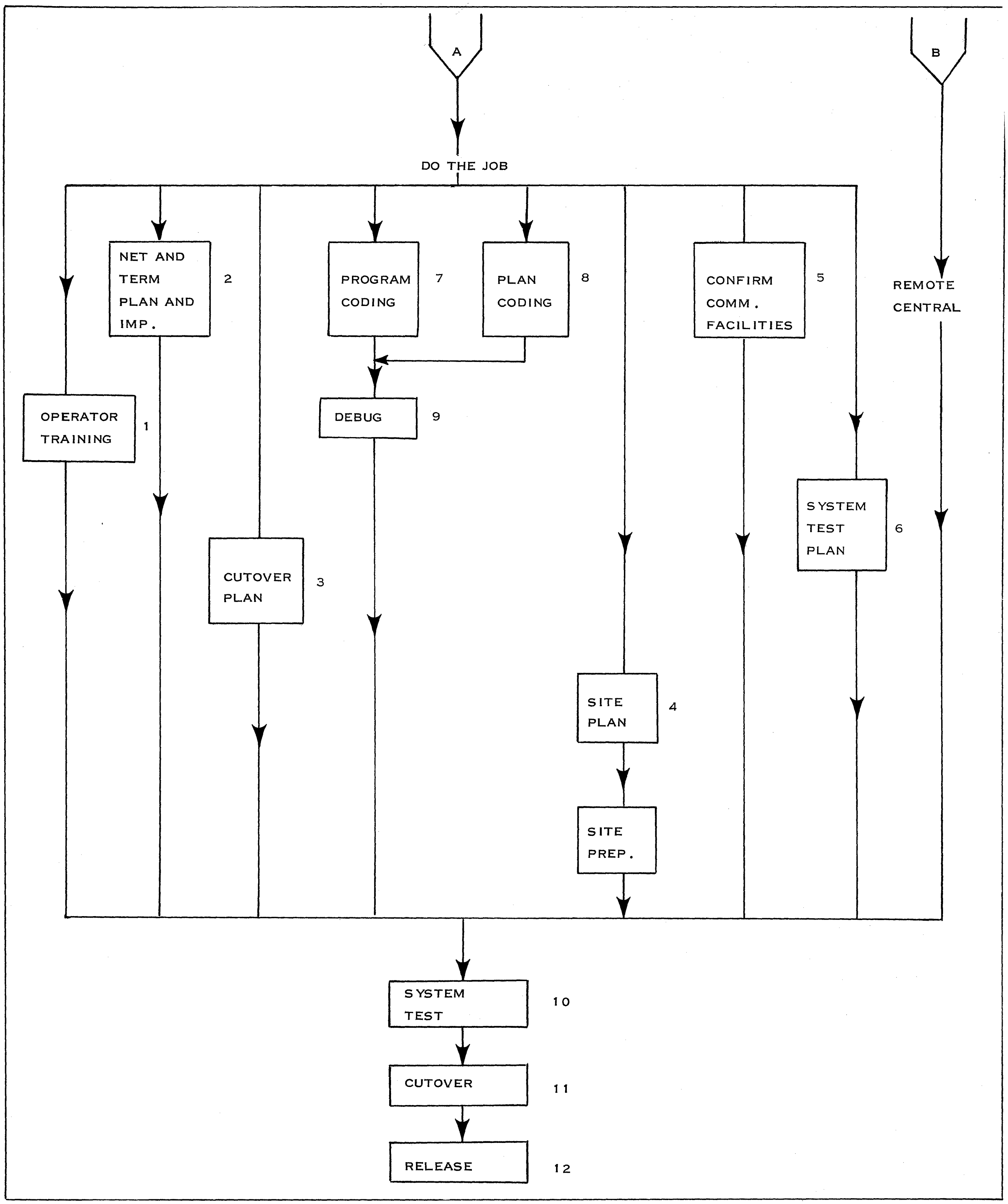


Figure 191. Implementation Chart: For Doing the Job from Initial Planning

adapters and wires or cables meeting IBM specifications purchased from local vendors.

If the PTT is involved, a meeting with the PTT must be held at least four to six months prior to the installation, with the primary purpose of settling all details necessary to permit the customer to order his communications facilities and to install his T/P system on schedule.

PTT involvement in a T/P installation would entail:

- One or a number of IBM T/P products to be attached to PTT facilities;
- The availability of PTT services that will enable communication between CPU's and terminals;
- System responsibility for the performance of the installation;
- Approval of the final installation when it meets the PTT requirements.

6. System Test Plan

The system test plan is necessary prior to going into the beginning of testing chains of unit programs that will ultimately end up in the total system test.

A system test plan usually considers the following:

a. Test in Chains of Unit Programs:
Actually, this is a continuation of the unit program test. This is where modules of the program would be combined to see if they interface with each other correctly.

b. Single-Thread Test:
For each message type in the system, we must make sure that the input message goes through the system and the proper output is initiated from it. During this test, we can test the errors due to improper input format, transmission errors, and hardware errors.

c. Multiple-Thread Test:
During this test, a number of messages will be entered into the system simultaneously to test the multiplexing operation of the hardware and the multi-programming capability of the programs.

d. Message Volume Test:
This is the test for finding out what the throughput capability of the system is. Here we test for file, overload, channel overloading, and buffer limitations. There is also a test to measure the system's processing capability.

d. Backup and Restart Procedure Testing:
This is the test for restart operations after a simulated failure has occurred in the test configuration. It is also a test for switchover operation. This is very important when a duplex system is concerned. It will also validate the switchover objectives. In other words, we will now find out how long it really takes to switch from a failed system to a standby system and complete the restart and recovery operations.

7. Program Coding

This is, naturally, one of the most important points, since it is the basis of the total system. This includes programming new applications and the conversion of existing programs.

Detailed flowcharts contribute over 75% of the undebugged coding effort and aid considerably in debugging. Flowcharting of major routines must be performed in parallel by the different members of the team and the coding must be performed in the same way.

Prior to the actual flowcharting and coding, the implementation team must review all the existing Type I and II programs that are available to see which may be used to accommodate the customer's application and system requirements. By choosing the proper support, a major reduction in the implementation effort can be realized. A major consideration is the installation date and the availability of a required program.

In the design phase, the design team may have based their core estimates and throughput evaluation on a packaged program that was intended to be announced prior to the customer installation date but whose schedule has slipped, and now requires increased core. Certain members of the team must therefore become versed in control programs for communications-based systems such as DOS with BTAM, OS with BTAM, OS with QTAM, CCAP, and other Type I and Type II programs.

When evaluating programming support, the advantages and disadvantages of each programming packages must be considered.

The advantages of using packages are:

a. More accurate estimate of core at proposal time;

b. Reduction of effort;

c. Reduction in time for implementation;

d. Assurance of program maintenance;

e. Proper documentation;

f. Useful sales tool;

g. Less dependent on individuals.

The disadvantages of using packages are:

a. Major modifications sometimes needed;

b. Not available soon enough;

c. Not available for some configurations (FDX);

d. Not efficient (restricted throughput);

e. Too big (because they cover all cases).

Programming packages can save two to four manyears, depending on customer requirements.

Even if a package program is used, certain additional control routines are required in a communications-based system, such as:

Additional Control Routines:

a. Good morning routine

b. Statistical routines

c. Message processing

d. Special terminal routines

e. Checkpoint

f. Restart and recovery

g. Message retrieval

h. Message intercept

i. Close-out routine

Most communications-control programming needs to reside in core, except for the majority of errror recovery programs. This puts more pressure on the coder to be efficient. Careful coding can save up to 25% of core.

8. Planned Debugging

This is also known as a program test plan. This consists of a schedule that considers items such as completion of program-module coding, availability of test facilities, availability of common carrier equipment, effects of RSDP's on program testing, availability of required programming systems packages, availability of utilities, and availability and knowledge of testing aids, additional user-written programs used to test system programs, assembly and testing of independent modules and, finally, the linking of modules and final test of programs before system test.

This consists of a number of sub-tasks, such as:

a. Conversion of Existing Programs:
There are basically three ways of converting from an existing system to a System/360:

- conversion programs
- simulation and emulation
- program rewriting

Even though simulators and emulators are used, there must be a complete rewrite of all programs for which there are no conversion programs. The simulators and emulators are used only to provide a smooth transition from the existing system to the System/360.

b. Hardware Availability:
For a communications system, common carrier equipment such as data sets and terminals will normally be required to check out the line control portion of the program. Therefore, this equipment needs to be planned for.

c. Additional User Programs:
Provision in the test schedule must be made for the writing of these programs. They are necessary for such things as simulating devices not available at the beginning of testing terminals, 2702's, RSDP's, etc.

d. Program Unit Testing:
In a CBS, Assembler language is the one most commonly used. The test translater (TESTRAN) can be used when programming in an OS/360 environment where Assembler language is used.

9. Debugging

This is also known as program testing. It is concerned with individual modules and linked modules which perform specific functions, such as message processing, header analysis or line control. System test, on the other hand, consists of linking together all the modules in order to bring messages into the system, process them, and route them to their destinations.

10. System Test

The system test is used to make sure that, in fact, the system meets its specifications, including throughput and error recovery under simulated operating conditions. The objectives of the system test are:

a. Determine if the system will operate as specified:
Here is where we make sure that all the unit programs link together properly, and we also determine if the program can handle multiple messages of various type mixes simultaneously.

b. Determine how and when the system will operate:
The system test enables us to predict at what time the system can, in fact, operate and also indicates where the problem areas are so that changes can be made.

c. Message throughput:
Through the system test we can find out how many messages in and out the system can handle to meet the peak hour requirements, and we can also find out what the maximum throughput capability of the system will be.

d. Obtain information for future improvements:
A system test enables the implementation team to pinpoint areas that can be changed to increase throughput and response. In addition, it could also show where savings can be made as far as core buffer is concerned, especially when using dynamic storage allocation.

11. Cutover

This is a very critical task. The success of a cutover depends on how well the system has been tested beforehand and the adequacy of the cutover plan. Cutover is defined as switching from an existing online system to the new IBM online system or, where the customer does not have an existing online system, the phasing in of new lines and terminals on an orderly basis.

12. Release

The system should be released for operational use when it has performed successfully throughout one operating period.

## 7. 3. DEVELOPING THE PRELIMINARY IMPLEMENTATION PLAN

In subsections 7. 1 and 7. 2 all the major tasks of the actual implementation have been described. An understanding of these tasks is necessary before a realistic implementation plan can be developed.

The preliminary implementation plan consists of a schedule and a prediction of manpower requirements, which would also include the assignment of tasks to specific individuals. The implementation concerns both IBM and the customer.

### 7. 3. 1 PRELIMINARY IMPLEMENTATION SCHEDULE

Prior to establishing the preliminary implementation schedule, the system designer should study an exhaustive implementation checklist and decide which tasks on the list are required for his particular system. Using the tasks from the check list he could then develop the schedule.

The following is offered as an implementation check list:

1. Form implementation team
2. Form implementation plan
3. Develop schedule
4. Assign tasks
5. Set up controls
6. Set up documentation
7. Review plan and schedule
8. Firm up functional specifications
9. Review system design
10. Train IBM and customer personnel
11. Central operating plan
12. Central backup plan
13. Remote operating plan
14. Remote backup plan
15. Review network design
16. Detailed system specifications
17. C. E. meeting
18. Freeze installation plan
19. Review programs
20. Conversion of existing programs
21. Program test plan
22. System test plan
23. Program unit testing
24. Single-thread tests
25. Multi-thread tests
26. Volume tests
27. Cutover plan
28. Central operator training
29. Plan central site
30. Central site preparation
31. Install central hardware
32. Plan remote sites
33. Remote site preparation
34. Install remote hardware
35. Plan remote training
36. Train remote operators
37. Remote physical interface
38. Remote logical interface
39. Common carrier meeting
40. Order facilities
41. Install facilities
42. Final system test
43. Cutover

44. Release

An additional list of factors affecting implementation planning is as follows:

1. Estimate in man months the time to program and debug

2. Assess manpower of the team based on their background and experience

3. Decide what courses are required

4. Obtain information on similar installations

5. Machine availability for testing

6. Obtain schedule from common carrier for implementation of network

7. Obtain schedules from Special Engineering

8. Make allowance for:

   a. Special meetings
   b. Travel time
   c. Writing routines to aid in debugging

A valid estimate for programming effort required is 4 to 5 debugged instructions per man, per day, from the beginning of the initial planning stage to the time the system is released. This includes every phase of the implementation effort. This is developed in detail in the pages that follow.

Over 50% of the total man effort required for the implementation is in the area of programming and system testing. If possible, obtain information from similar installations to help estimate the man-effort required. Machine availability and package program availability must be verified and included in the schedule. Concurrence must be obtained from the common carrier to assure installation of hardware by the proposed dates. Make sure that Special Engineering is on board and that they have realistic RSDP schedules to meet your testing and installation dates. Naturally, time should be allowed for such things as travel time from the customer site to a testing location.

*Typical average time and manpower estimates for the communications portion of a small CBS using DOS/BTAM or OS/BTAM is as follows:

a. A system that has existing lines and terminals would require 2 to 3 men - six months to one year.

b. A system that does not have existing lines and terminals would require 3 to 4 men - 6 months to one year.

The total number of men in the installation team includes both the customer and IBM personnel.

A schedule for implementation is based on the number of debugged instructions per man, per day. The number of instructions is spread over the period from the initial planning to the time the system is operating. This includes all tasks pertaining to the implementation effort as described in sections 7. 1. 0 and 7. 2. 0.

The time it takes for the implementation effort includes a constant for education time for the total implementation team, plus a variable for programming, testing, and the other major tasks associated with programming testing, and a constant for cutover.

The rules of thumb for estimating time for implementation are:

a. Education time: 1 to 2 months

b. Time to write and test for debugged instructions written in Assembler language:

   1. CBS programs: 2-10 instructions/day (average of 4)
   2. Tables: 25-50 entries/day (average of 40)
   3. Offline programs (background): 6-15 instructions/day (average of 10)

c. Time for cutover: 1 to 1. 5 months

For example, if it has been estimated for a hypothetical system that it would take 4K instructions for T/P, 1K for table entries, and 20K for instructions for offline programs, the total estimated time for the installation, based on a programming team of five men, would be:

$$ {}^{**}T = \frac{1}{(\text{NO. OF MEN})}\left(\frac{\text{NO. OF T/P INST.}}{4} + \frac{\text{NO. OF TABLE ENTRIES}}{40} + \frac{\text{NO. OF OFFLINE INST.}}{10}\right) + 44 $$

*These times are for planning purposes only.

** The constant, 44, represents the additional two months required for education and cutover.

$$T = \frac{1}{5} \left( \frac{4000}{4} + \frac{1000}{40} + \frac{20000}{10} \right) + 44$$

$$= \frac{1}{5} (1000 + 25 + 2000) + 44$$

$$= \frac{3025}{5} = 649 \text{ PROJECT DAYS}$$

The approximate numbers of years required would be:

$$\text{Years} = \frac{649}{12 \text{ x } 22 \text{ days/month}} = 2.5 \text{ years}$$

It can be concluded that for this example the total implementation effort would take 2. 5 years. An additional man is required for the hardware and communications area, which would bring the installation team to a total of six men.

The communications portion of the implementation would take approximately three programmers.

$$T = \frac{1}{3} \left( \frac{4,000}{4} \right) + 44$$

$$T = 333 + 44 = 377 \text{ project days}$$

$$= 1.4 \text{ years.}$$

An additional man is required for hardware and the communications area. Therefore, the total implementation effort for the communications portion of the system would take four men 1. 4 years. It is noted here that this is only an example and that 4K instructions is a large number for a CBS with a simple application

An example of a schedule that has been met for a Tele-processing system is shown in Figure 192. The total effort took five men a period of 15 months. This does not include such things as travel time, meetings, vacations, illness, etc. It also excludes the addition of specialists, when necessary, to aid in the installation. This is a schedule that has been met and is for a system that did not use any IBM package programs, but consisted of a 4-5K instruction CBS control and application program geared to customer needs. This system consisted of a large number of terminals and lines, but the CBS application was simple.

### 7. 3. 2 ASSIGNMENT OF MANPOWER

The system designer at the preliminary implementation plan phase must be aware of the kind of background required for the implementation team so that he can plan for proper education and proper assignment of tasks in order to determine the number of men needed for the implementation.

The skills required for the implementation team are:

1. Knowledge of communications

2. Experience, or education, in programming and implementation of T/P systems

3. Programming experience

4. Knowledge of System/360 direct access devices and file organization

5. Binary systems

6. Knowledge of the customer's specifications

These skills are identical to those required for system design, with the exception of having someone who has been involved in the implementation of a CBS or at least a batch system.

An example of the allocation of jobs for a CBS (T/P system) performing message control and sales order entry with an implementation team made up of four men, one coordinator, two programmers, and one man concerned with the remote equipment and the network is as follows:

1. Three men at the central system:

   A. One coordinator

   B. Two programmers

   C. Prime tasks assigned to this group:

      1. Education

      2. Investigation of program package

      3. System specifications

      4. Programming and coding

      5. Plan testing and debugging

      6. Testing and debugging

      7. Operator procedures

      8. Backup procedures

      9. System test

2. One man on remote equipment and network

A. Prime tasks assigned:

1. Education

2. Network design

3. Cutover planning

4. Training of Operators

| TASK | NO. OF MEN | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | MAN MONTHS |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| INITIAL PLAN | 2 | ■ | | | | | | | | | | | | | | | 2 |
| PRELIMINARY EDUCATION | 4 | | ■ | | | | | | | | | | | | | | 4 |
| EXEC. SEMINAR AND PROG. PACKAGE | 8 | | | ■ | | | | | | | | | | | | | 4 |
| NETWORK DESIGN | 1 | | | ■ | ■ | | | | | | | | | | | | 1 |
| FILE COURSE | 4 | | | | ■ | | | | | | | | | | | | 2 |
| GENERAL SYSTEM DISCUSSION | 5 | | | | | ■ | | | | | | | | | | | 5 |
| PROGRAMMING SCHOOL | 4 | | | | | | ■ | | | | | | | | | | 3 |
| PROGRAM PACKAGE DETAIL | 4 | | | | | | ■ | ■ | | | | | | | | | 2 |
| SITE DEFINITION | 2 | | | | | | ■ | | | | | | | | | | 2 |
| PRELIMINARY CUTOVER | 2 | | | | | | ■ | | | | | | | | | | 2 |
| SYSTEM SPECIFICATION | 4 | | | | | | | ■ | | | | | | | | | 3 |
| SPECIFIC NETWORK DESIGN | 1 | | | | | | | ■ | | | | | | | | | 1 |
| PROGRAMMING AND CODING | 3 | | | | | | | | ■ | ■ | ■ | ■ | ■ | | | | 15 |
| PLAN TEST AND DEBUG | 1 | | | | | | | | ■ | | | | | | | | 3/4 |
| UTILITY ROUTINE | 1 | | | | | | | | ■ | ■ | | | | | | | 3/4 |
| SITE PLAN | 1 | | | | | | | | | ■ | | | | | | | 1 |
| OPERATING PROCEDURES | 1 | | | | | | | | ■ | | | | | | | | 1 |
| TEST AND DEBUG | 3 | | | | | | | | | ■ | ■ | ■ | ■ | ■ | | | 15 |
| PLAN CUTOVER | 1 | | | | | | | | | | ■ | | | | | | 1 |
| TRAIN OPERATOR | 2 | | | | | | | | | | | ■ | ■ | | | | 4 |
| SITE PREPARATION | 2 | | | | | | | | | | | | ■ | ■ | | | 4 |
| CONSOLE PROCEDURES | 1 | | | | | | | | | | | | | ■ | | | 1 |
| INSTALL CPU | 4 | | | | | | | | | | | | | | ■ | | 1 |
| TRAIN CONSOLE OPERATORS | 1 | | | | | | | | | | | | | | ■ | | 1/4 |
| NETWORK CUTOVER | 5 | | | | | | | | | | | | | | ■ | ■ | 5 |
| DOCUMENT CLEANUP | 5 | | | | | | | | | | | | | | | ■ | 5 |
| TOTAL MAN MONTHS | | | | | | | | | | | | | | | | | 88 3/4 |

Figure 192. Example of an Implementation Schedule

## 7.4 CONCLUSION

The implementation effort and plan are as important as -- in fact, more important than -- the design phase. Careful consideration must be given to the implementation plan, schedule, and manpower requirements at proposal time.

The implementation effort, if not manned and planned properly by IBM and the customer, will lead to unexpected expense to both. The salesman and S. E. must be aware of the implementation effort requirements for a small CBS in order to advise the customer to prepare his implementation team and to ensure that the implementation will be on schedule.

This section of the Handbook is offered as an aid to the system designer in the area of implementation planning to enable him to develop a realistic plan and approach to the implementation problem.

# INDEX

Z10-0003-0

Printed in U. S. A.

Z10-0003-0

IBM

IBM World Trade Corporation
821 United Nations Plaza
New York, New York 10017