1970

A study of the use of a small computer as a communication interface for a large on-line system

Robert Jay Schwartz Jr.
Lehigh University

A Study of the
Use of a Small Computer as a
Communication Interface for
a Large On-Line System

by

Robert Jay Schwartz, Jr.

A THESIS

Presented to the Graduate Committee

of Lehigh University

In Candicacy for the Degree of

Master of Science

in

Information Science

Lehigh University

1970

******

ACCEPTANCE

This thesis is accepted and approved in partial fullfilment of the requirements for the degree of Master of Science.

Sept 10, 1970
Date

Arthur I. Larky
Dr. Arthur I. Larky
Professor in Charge

Donald J. Hillman
Dr. Donald Hillman
Chairman of the
Department of Philosophy

ACKNOWLEDGMENT

During the research for this thesis I received valuable assistance from Mr. J.E. Tilley, Miss Jean Lay and Mr. Thomas Maile of Pan American World Airways in New York City. Mr. Albert Siegel, Manager of the Lehigh University Computing Center, has been a constant source of advice and encouragement. I am deeply indebted to Mrs. Joan Jones who provided valuable typing and proofreading services.

A considerable amount of the information contained in this thesis was originally developed by the General Dynamics Corporation of San Diego, California under contract to Pan American.

Pan American has agreed that information derived from their documents may be released. The letter from Mr. J.E. Tilley, Manager - Forward Planning that authorizes this release is available in the author's files.

TABLE OF CONTENTS

TABLE OF FIGURES
Appendix III

ABSTRACT

Increasing use of Digital Computers in communication environments has emphasized the need for an efficient method of interfacing communication lines to the Central Computer. Hard-wired multiplexors, acting as slaves to the Central Computer, must look to it for instruction for polling, error checking and other tasks. This has slowed the Central Computer from its main tasks of referencing data and constructing replies.

The advent of mini and small computers has provided an alternative. A programmable small scale computer can handle:

1. Initialization and repeated polling for Terminal Interchanges.
2. Acceptance and routing of messages resulting from 1.
3. Routing of messages to Terminal Interchanges.
4. Real-time statistics and error logging.
5. Real-time error detection.
6. Output system status parameters to the operator.
7. Perform off-line diagnostics and utility functions.

This thesis discusses the design of such a Communication Interface System in an airline environment. Pan American World Airways, Inc. is chosen as the model for this discussion, although the plans described herein are not intended to represent those of Pan American.

An analysis of the requirements in terms of data load and speed are a critical part in determining hardware specifications. After all hardware needs are determined, a practical configuration is specified. An outline of the software necessary to operate the Communication

Interface System is then developed. Finally, an analysis of various small computers commercially available shows the system to be workable and practical.

Hence, the "Study of the Use of a Small Computer as a Communication Interface for a Large On-Line System" develops the design of a workable Communication Interface System that can interface with a large computing facility and many duplexed medium - high speed communication lines. A currently operational system in New York at the Chase Manhattan Bank is cited. Interface units or programmable Communication Front-Ends will become increasingly evident as Communication requirements proliferate.

Maintaining communication between large-scale on-line computer systems and numerous, diversified remote entry terminals is a task of major consequence. The central computing facilities cannot interface directly with incoming telephone and dedicated private wires, and hardwired communication front-ends are time-consuming, expensive and inefficient. Current developments indicate the advisability of a programmable medium-size computer to serve as an interface. This thesis will discuss a design and implementation of such a Communication Interface System (CIS).

The airline industry has a major commitment to third generation real-time reservation systems. Current research in this area, notably at Pan American World Airways, form the basis for this presentation. Pan Am, which in the following text will simply be referred to as The Airline, is installing two IBM-360 Model 65s in an on-line totally redundant configuration.[1] This system, known as PANAMAC/360, is shown in Figure 1 and henceforth will be referred to as the central computing facility or CCF. It will be noted that the CCF is defined as encompassing the Central Processing Units, discs, drums, tapes, and all other associated peripherals and interfaces between those peripherals and the CPU. Therefore the CCF stops where the communication data is to enter, and a system is needed to present incoming data in a logical, efficient and fast manner. IBM S/360 cannot interface directly with communication lines due to its basic construction. S/360 does provide two types of input channels for

communication with peripherals or external systems: one multiplexor and a variable number of selector channels. An IBM 2703 Transmission Control Unit (hardwired front-end) can be attached to multiplexor channel as shown inFigure 1. The multiplexor will operate in either burst or byte mode, allowing it to interface with numerous slow-speed peripheral units or one high speed unit.

Selector channels operate only in burst mode and are used by one high speed unit at a time. No byte interleaving can take place on the selector channel due to the extremely fast transfer rate. IBM's Standard Interface concept allows any channel to communicate with any peripheral through an appropriate interface unit. The CIS will interface through a data processor coupler to the 360 multiplexor channel and operate in non-burst mode.

Two alternatives are open in attempting to interface outside lines with the CCF. Firstly, a hardwired front-end (communication interface) can serve as the interface. A non-programmable unit such as this must look to the CCF for its instructions and must rely on the CCF for storage facilities. Effectively, the unit acts simply as a buffer and interface. This reliance on the CPU of the CCF uses unnecessary processor time and core storage, and slows the CCF from acting upon its main problem: decoding messages, referencing and accessing the appropriate table or data, and constructing a reply. Also, expansion capabilities dictate the use of incoming lines for other purposes than solely reservation needs. Future communication

links to other Central Processing Units will include Flight and Weather information on a real-time basis, to provide updated flight plans. This system, using redundant IBM 360/40 computers is currently operating and producing new routes on heretofore non-plotted sectors. Cargo sales and records as well as maintenance files and crew records will eventually be entered through the system to their respective systems. Also under design by the airline industry is an Automated Fare Quote System. Through use of a central large-size computer an agent of any airline will be able to access instantly or on a delayed basis any possible fare. Use will occur when an unusual routing occurs and the lowest cost is desired through combination of various types of fares (e. g. excursion, affinity group, standard, etc.). Hence the Communications Interface System will interface to this system too. The Airline Industry also maintains a switching center for interline messages called ESS (Electronic Switching System) where various airlines can query each other on matters such as space availability. Provision must be included for ESS as well. Another new development is automated ticketing. A passenger can query a computer terminal for availability and purchase a ticket through use of a credit card. Presently a limited trial is being run at Chicago's O'Hare Airport where passengers may purchase a ticket for a flight to New York if they hold a special magnetic-encoded credit card. Eventual expansion to a full system will dictate use of a separate central computer to handle space checking, fare

accessing and ticket issuing. Interface capabilities are also needed here.

Consequently, a number of hardwired front ends would be required to handle the data and the economic justification as well as the technical advantages of the more expensive medium-size computer is indicated. It has therefore been determined that the only feasible interface system will be an independent programmable medium size computer. The Communication Interface System, referred to as CIS, must be a total hardware and software system capable of supporting a multiple CPU CCF. The selection of a programmable front-end allows many features and options unavailable or impractical with the non-programmable hardware. Objectives of the CIS are to perform terminal polling, message switching, store (spooling to disk) and forwarding, buffering, error detection, network status reports, and statistical reporting and error diagnostic logging. This will provide for the efficient utilization and flexibility of communication lines and central computing facilities.

Reservation systems with prompt response have become accepted by the public, and the airline industry relies on their systems almost totally for booking revenue passengers. Also, other previously mentioned systems such as flight planning and cargo manipulation are critically needed on an on-line basis. Hence systems such as the CCF are duplicated for high backup capability. The Communication Interface system must therefore employ as much cross connection as

possible to approach a truly polymorphic complex. Eventual system design must incorporate duplication of front-end units, such as the CPU, tape drives and so forth.

The problem has now been briefly outlined. The next step is to specify a general system configuration and then to develop hardware and software requirements. Through an information flow analysis, hardware specifications will be shown to be realistic and functional. Finally an analysis of the applicability of various commercial medium-size computers will be shown.

HARDWARE DEVELOPMENT

Figure 2 details the eventual CIS configuration. Reference to it now and continuing throughout this section will provide the reader with a guide as hardware requirements are developed. Considerations for hardware choice should encompass the following features:

1. Operator Control Console (OCC)
2. Real-Time interrupt clock
3. Direct memory access availability and transfer rate
4. Memory cycle time
5. Memory word size
6. Core size
7. Index and general registers
8. Tape unit functions, speed and density
9. Disc Size
10. Interrupts
11. Instruction set
12. Restart ability

Eventual goals are to be able to interface with 31 fully duplexed high-speed lines at 2000 Bits Per Second, one 2400 BPS fully duplexed ESS message line and two 75 Baud lines for read-only teletypes. Each communication line is capable of handling up to 10 terminal interchange units. Each terminal interchange unit can be responsible for up to 30 agent sets. In this system approximately ten of the 32 lines will be for reservation agent sets and ten more will be allocated for cargo agent sets. The remaining twelve lines will be available for expansion. One terminal interchange will service each line and ten agent sets will be attached to each interchange. The two read-only teletypes are used for operator messages from the CCF Reservation and Cargo Systems.[2] As well as interfacing with agent sets, teletypes and CRT sets (see Figure 3), the Communication Interface System must interact with the other systems described earlier.

Operator Control Console

The operator control console should have a keyboard type entry unit with a hard copy record of input and system messages available. Although not essential, a CRT can provide intermediate messages of a transient type to the operator. These messages may be recorded onto tape (magnetic) or TTY for later analysis if desired. Hence we choose a CRT and a KSR TTY. The input keyboard and output devices should have twenty-six alpha characters, ten numerics and numerous special and punctuation characters. The special keys should include frequently used functions such as insert, delete and replace. The CRT display matrix should be 1024 characters.

Clock

Since functions such as polling must be carried out in very small increments of time, a real-time interrupt clock is necessary. Assume t is the basic time increment cycle of the clock. Then it will generate an interrupt every 1/(nt) seconds, where n is initialized from the operator control console or under program control. The clock may be used to produce time information which may be made available in a 16-bit register for reference. Since the date and time are to be loaded at initialization time, the date, Julian or otherwise, can be incremented by the clock. The time information will provide for time-stamping incoming and outgoing messages and act upon time-initiated messages at the proper time.

Machine Speed and Utilization

In Appendix I a determination that the average agent set transmits one 96 character block/minute is made. Assuming a 1 u sec cycle time computer, the calculation which follows will show the workability of this system.

At ten agent sets/terminal interchange and one terminal interchange/line, each line transmits 960 characters/minute or 16 characters/second. If each of the 20 line transmits continuously, a 320 char/sec or 1920 bits/second (6 bits = 1 char) will be received at the CIS. This corresponds to one character every 3 milliseconds. It has been estimated by users familiar with the DDP-516 that processing a single character into core buffer will require about 75 u sec; thus, 2.925 ms are left for the remainder of the CIS work. Therefore I/O can be estimated at 2.5% (75/3000) of the available machine time, for a system with 200 agent sets.

The characters must be transferred to the CCF for final processing. This is, however, done under Direct Memory Access, and would add only 1 memory cycle (1 u sec) per character. Thus the total I/O transfer time overhead remains on the order of 75 u sec/character.

Direct Memory Access

Direct Memory Access is the only solution for the amount of data to be transferred. The DPC (Data Processor Coupler) should permit transmission in both directions during alternate DMA cycles. It should also generate parity bits in transmission to the CPU and check parity in reception from the CPU. Four channels with 80KC transfer rate at burst mode are needed.[3]

An address counter and range counter for each channel are necessary. Instead of transmitting DMA in bits, data will be accumulated into blocks of approximately 100 characters and then sent in parallel.[4] The MDMA (Multiplexed Director Memory Access) should provide temporary storage for 64 2-character words and 128 buffer addresses at 2/channel.

Word Size

A core size of 32K 16-bit words should provide space for programs, operating system programs and operating data. In cases of core of this size indirect addressing usually provides the means of addressing all of core. The CPU must provide suitable registers and capability for indirect addressing. In addition, two general (A & B) registers and necessary counter and overflow registers are indicated.

Tape Drive

Standard Magnetic Tape drive functions should be:

1. Write EOF
2. Read Record
3. Write Record
4. Backspace Record
5. Rewind Record
6. Unload

The read feature can be used to skip a record or file, eliminating the need for these special commands. Seven or nine track is currently used in the industry but since the trend is towards nine track tape, the handler should have nine track read/write ability. Eight hundred bits per inch is the most compatable with present industry products so therefore the drive must at a minimum have that feature.

Disk Drive

The disk drive, as noted previously, will act as an extensive buffer and contain system and operating programs. During heavy loading of the S/360 CPUs the disk will store data. To calculate disk size we first note that a core full of programs = 16 x 32K = 512K bits. Since only a relatively small part of the disk is necessary for storage of programs an estimate of 3 x 10 characters seems adequate. It should be noted that since the system is fully redundant, a second disk will be available for access. A multiple moveable head unit with detachable disc packs will provide flexibility.

Interrupts

Interrupts, the ability to respond to some external signal, is an integral and critical part of this Communication Interface System. If we are to handle 32 communication lines, then at least 48 channel or priority interrupts are needed. The extra sixteen interrupts will provide for the various miscellaneous peripheral devices. When an Input/Output interrupt occurs the general registers are saved. By saving a status word at the commencment of an interrupt, return can be accomplished to the previous program when the interrupt program has been executed. The interrupt masks will contain the address or sector of the appropriate program or subroutine to be executed. A single interrupt line is used to handle signals generated by the real-time clock (decrementing counter), Operator Control Console or generated overflows. The CIS must poll each of these sources to determine the origin of the interrupt.

After determination, handling precedes in the pre-defined manner. Various uses of interrupts served by the Interrupt Processor Module include signalling when the primary input buffer is full and initiating transfer to the secondary buffer. Also, when a message transfer has been completed an interrupt will be generated.

Instruction Set

The instruction set is divided into seven areas listed below. Although a small computer can be programmed using only a few commands, that method would require excess coding and machine time. Hence we shall attempt to include the necessary instructions in the hardwired set.

I. Load and Store
   a. Clear A
   b. Load
   c. Store A
   d. Load Index
   e. Store Index

II. Arithmetic
   a. Add memory to A, result in A
   b. Subtract memory from A result in A
   c. Increment A
   d. Negate A register
   e. Increment memory address

III. Logical
   a. Logical "And"
   b. Exclusive "Or"
   c. Complement A
   d. Complement Sign

IV. Shifts
   a. Left Shift A
   b. Right Shift A
   c. Left Rotate A
   d. Right Rotate A

V. Control
   a. Unconditional jump
   b. Unconditional skip as a register function (test and Branch)
   c. Eight or more sense switch skips
   d. Enable and inhibit interrupts

VI. Input/Output
   a. Output control without peripheral request
   b. Output control with peripheral request
   c. Input control with peripheral request
   d. Skip on peripheral sense line

Restart Capabilities

In event of a power failure core should not be reset or altered and the registers should be saved. Conversely the re-initialization of power to the CPU should not reset core without specific instruction. These requirements will enable the CIS, through the use of checkpoints, to re-start in the event of a power loss. Since the total installation, including the Central Computing Facility, is supported by emergency generators, total lost messages would be kept to a reasonable amount.

Hardware, therefore, has been specified that will satisfy all projected demands of the Communication Interface System. The next step is to design software capable of managing the system.

SOFTWARE SPECIFICATIONS

This section will deal with the various aspects of Software needed to support the Communication Interface System. The topics are:

1. Programming Language
2. System Initialization
3. Utility Programs
4. Message Switching
5. Terminal Interchange Line Control
6. Methods of polling and responses to error conditions
7. Miscellaneous error recovery procedures
8. Diagnostic analysis - on-line and off-line

Individual problem programs will not be detailed, but rather the specifications they must handle will be discussed.

Programming Language

Choice of a programming language is dictated by the following considerations: real-time environment, massive I/O transfer, highly repetitious cycling of programs and reasonable limitations on core size. The choice, which is between a combination of assembly and high-level languages or solely assembly language is not difficult to make. Supervisory programs will necessarily be written in the assembly language of the machine or an efficient code generating high level language if available. A low-level language will also reduce time, increase I/O transfer and facilitate linkage if used throughout. Repeated use of routines will easily undercut the additional programming time required for assembly versus high-level language. Therefore, only an assembler is needed, and this may be either the manufacturer's or a user written version. Most small computers now are available with at least an assembler.[5][6]

Figure 4 shows the major program sections while Figure 5 outlines the control group.

Initialization

The Initialization Module will require at start time the following entries which must be alterable during system operation as well:

1. List of lines to be polled and characteristics
2. Data Buffer Size
3. Diagnostic Logging Volume:
   amount of diagnostic data to be provided. To vary from none to real-time response.
4. Recovery file - create or not
5. Time limit for response of each communication line
6. Month, day, hour, minute, second.
7. Hard copy of diagnostics desired - yes or no

The module should have default parameters for the above entries.

Creation of the recovery file on disc will enable the System Program Loader to provide instant re-start for the system. The entries provided at initialization time as well as any default parameters assumed will be loaded on the recovery file. The Initialization Module is to be part of the utility set that is loaded by the System Program Loader (SPL). All programs will be loaded by the SPL except utilities such as Core Dump and Core Flush routines (described below) as well as the SPL itself, all of which will be loaded by the Bootstrap File. This file will be loaded by a switch on the Operator Control Console. The SPL will load from either tape or disc. Although disc will be used during normal cycling of the system, tape loading is useful during diagnosis and testing of the system, and will be available.

Utility Routines

Due to the specialized needs of this system and the general low quality of manufacturer supplied utility programs, the user should undertake to write his own wherever possible.

The Core Dump Routine should write in an output directly compatible with S/360 printers, as these will be used for displaying any dumps of major consequence. Minor dumps will be sent to the TTY or CRT at the OCC. Core Dump output should also be compatible with binary formatting required for reloading through the SPL. The Core Flush Routine is simply a program capable of loading any or all memory word values with any pattern. Normal usage would be flushing memory with all zeroes or ones to facilitate reading of core dumps or debugging.

Non-Periodic Tasks

Non-periodic tasks are handled on a demand-only basis by the Non-Periodic Executive. When interrupts or periodic tasks are not being performed, the Non-Periodic executive will cycle and the I/O Processor determine the status of the following message queues:

1. CCF Queues
2. High-Speed (Reservation, Cargo, etc.) Data Line Queue
3. ESS Data Line Queues
4. Peripheral

The message queues will contain the number of messages to be processed, as well as the beginning and ending addresses.

The CCF queue is a table for incoming messages to the various CCF's.

The High-Speed Data Line Queue contains a table of messages for the various Reservation, Cargo, fare quote, etc. lines that are being serviced. Similarly the peripheral queue holds the lists of data for: Read-only TTY, OCC CRT and keyboard, as well as the magnetic tape and disc units.

Linkage and Management

The Program Linkage Routine holds responsibility for transference of control between application and control programs as well as within separate application programs. The Space Manager contains two subsets; core management and disc management routines. These maintain efficient allocation of core and disc storage based on the dynamic needs of the system. They supervise all attachment and release of large blocks of core and disc space.

Parity Checking

Message switching will be a subset of the I/O Processor. It will accept messages from each CPU, interpret the header and route to the proper queue for distribution to the proper line. Likewise, it will receive messages from the communication lines and route to the correct Central Processor queue. Actions in both directions will take place with a minimum of delay and no modification of the text. Pre-checking of parity will be performed by this routine by examination of the message header ID. (See Figure 6.) The Message Output Routine, also a subset of the I/O Processor, checks the queues for output and initiates it. The interrupt processor determines when the entire message has been transmitted. Whenever an invalid message or block is detected it will be

purged in the following manner: The address assigned to it will be made available for use by another message. All messages received from the CPUs must have a header ID, text and End of Message character. Messages received will have the Cyclic Check Character (CCC) as the first bit of the Terminal Interchange Address and last bit of the EOM (End of Message) designator. A CCC will be generated for all communication to and from the terminal interchanges. A faulty CCC will cause either unit to disregard the message segment and request retransmission. A Cyclic Check Character is accumulated by adding each incoming character to the previous one, and carrying overflow bits into the least significant place.

Line Control

Line control will actually be controlled by the CIS as it will control functions to the terminal interchange units and accept status and functions from those units when accepting messages. Line control will be separate and independent from the CPUs excepting for the interpretation of routing indicators contained in message headers.

Polling: Direct and Hub

Terminal Interchanges out in the field will serve to combine up to 30 agent sets each for transmittal on one line. These lines will require periodic polling routines to purge T-I held messages to the CIS. Figure 7 describes the interconnection for polling purposes of the T-Is on a communication line. Two types of polling will be used: Direct and Hub. In hub polling, the CIS interrogates the furthest terminal interchange over the respective output line. That interchange should respond with

all completed messages and transmit a "go-ahead" to the next interchange address. The next interchange then transmits its information and sends a "go-ahead". This process is repeated towards the CIS until the line has been completely polled and the "go-ahead" is sent to CIS by the nearest interchange. In both cases if no message is ready, the "go-ahead" signal is given immediately. The CIS can also be viewed as a terminal interchange when another data communication system is referencing it, and thus can serve as another station (somewhat more elaborate) in the polling chain.

Polling by either method is concurrent. Once a poll of a communication line has been initiated, another line is polled. This method efficiently utilizes the high speed of the CIS CPU and prevents idling during relatively slow speed polls. Direct polling is simply the process of specifically addressing a terminal interchange from the CIS.

Polling: Error Conditions

Error conditions will arise with a greater frequency here than in other parts of the system because of the use of external communication lines.

Seven possible types of error condition on polling are possible and can be expected at times in an actual environment.

1. Terminal interchange does not respond or responds improperly to hub or direct poll.
2. Terminal interchange does not respond or responds improperly to hub.
3. Terminal interchange transmits invalid next interchange address with a "go-ahead".
4. Carrier line failure.
5. Input line lock.
6. Output line lock.
7. Parity error.

When a unit fails to respond to a hub poll the previous terminal interchange is polled again. After three successive hub failures, three attempts will be made to directly poll the unit. In either case three direct failures will cause the terminal interchange to be skipped. In the case of failure during normal direct polling, three further direct attempts will be made. In the case of any failures to respond, error conditions are noted on the diagnostic tape. An attempt will be made during each cyling of the system to poll the interchange unit until it responds or is dropped from the polling pattern by the operator through the OCC. If the unit does respond eventually, error indicators are turned off and its response is noted on the diagnostic tape. The next Interchange Address (NIA) can be reset by the operator so that a combination of direct and hub polling will take place.

Normally the NIA is preset at a terminal interchange to the next interchange unit. If an invalid NIA is detected by the CIS it

will first time out the line to see if another terminal interchange responds. If not, it will attempt to change the NIA in the erroneous interchange unit. If it responds properly, the error indicators are turned off. If the unit still sends a bad NIA it is dropped from the polling sequence and the previous terminal unit is given a new NIA to branch around it. The operator is appraised of the situation and has intervention and modification capacity.

Error Recovery Management.

The Error Recovery Management module handles the following:

1. Checkpoint marking - keeping track of addresses, register contents, device status and system status at prescribed intervals.

2. Attempts to rerun recoverable errors.

3. Logging of error on tape or DASD

4. Visual indication to operator of non-recoverable errors.

5. Removing frequently failing lines and terminals from systems.

In the event of failure of the CPU of the Central Computing Facility the CIS must attend to various tasks. It should notify the ESS computer interface that it will not accept messages until further indication. This will prevent addressing of the airline's office function designator and the other airline's computers will receive a down message. The CIS will clear all completed messages in its

queueing areas that are waiting to be sent out. Transmission of a message to all agent sets through their terminal interchange will inform agents of machine status, and that no further messages will be accepted. Upon correction of the downage all-clear messages will be sent. These procedures will provide for a smooth shut-down of a failing system instead of catastrophic failure. In the event of non-catastrophic failure of the CIS CPU, the system will attempt a switch to the alternate unit. Catastrophic failure (indicated by a lack of CIS messages and/or a diagnostic from the CCF) will necessitate operator action in switching units. Through use of check-points restart will be facilitated and the system's lost data will be minimized. Failure of a peripheral will be noted by the CIS and request operator intervention.

Parity Errors

Various parity errors will be detected by the CIS. The CIS will not delete messages that it detects have incorrect parity, but will flag the status word and thereby notify the CCF. These cases are: CCF to CIS, ESS to CIS and terminal interchanges to CIS. If ESS or CCF detects a bad message from the CIS it will notify to that effect and request transmittal. In cases of repeated failure of the CIS CPU the operator will be notified and may take appropriate action that includes switching to the alternate CIS CPU.

In the event of an attempt by the CCF CPU to overload the CIS

with a transfer rate excessive for the core resident queue or communication lines, the overage can be backed up on CCF disk or directed to the CIS Disk. Assigning a maximum queue size will prevent domination of the CIS by a runaway system. Priority can be established for messages at a previous time that will enable queueing of low priority messages (e.g. weather data) during heavy input/output loads.

A carrier-on failure may occur when noise bursts appear or a carrier drops on a line. Eight successive errors of this type will cause the CIS to lock out that line and not receive until a new syncronization pattern is determined. If the carrier-on failure is not constant the CIS will timeout the line and attempt to repoll. Input or output lines may lock on occasion, and these conditions are relatively easy to determine. If the CIS receives a message greater than a pre-determined length without an End of Message delineator the CCF will be re-queried. Upon a second sequential occurance the CIS operator will receive a priority message at the OCC. Upon direction from the operator the CIS will repoll the CCF. Similarly, if an input line locks on with continuous spurious data the CIS will determine by lack of the EOM at the proper interval. After a second poll the line will be eliminated from the polling sequence and the operator will be notified.

In all cases the operator has discretionary power to re-initiate or drop procedures. Counters will be kept of total errors

and total unresolved errors of each type. That is, if an error has been corrected the unresolved counter will be decremented. Status will be available at the OCC and all may be re-initialized through the OCC. Besides the function of error logging, a data log will be kept on normal functions such as message rate, queue sizes, traffic figures on utilization by various terminal interchanges and Central Computing Facilities.

Simulation

It should be noted that simulation is a valuable tool to test the effecacy of the system before it is put on-line. Testing is not practical in the real on-line environment because of the critical nature of the real-time reservation system and other CCFs. Hence simulation holds the key to development and testing. A computer, with a tape simulating communication messages and problems, can interface in the stead of a communication line. The testing computer should also simulate the conditions of operation looking at the CCF. The operating system such as IPARS[7] running under DOS, that the CCF will use, should also be simulated. Figure 8 shows a sample configuration for testing. Various effects and changes that can be noted include:

1. Response time if number of TI/line increased
2. Changing polling techniques
3. Byte vs. Burst mode for system inter-communication
4. Message rate by adding agent sets
5. Core vs. disk as overflow buffer
6. Peak system utilization and growth potential

Most important, debugging of unforeseen factors may take place through simulation.

Summary

To summarize, the CIS control package will accomplish the following functions:

1. Initialize and Repeat Polling for Terminal Interchange.
2. Accept and Route Messages resulting from 1.
3. Accept and Route Messages from the ESS lines.
4. Route Messages addressed to the Remote Teletypes.
5. Perform Real-Time Statistics and Data Logging.
6. Perform Real-Time Error Detection.
7. Output System Status Parameters to the OCC and Receive Instructions from the OCC.
8. Perform Off-Line Diagnostics and Utility Functions.
9. Accept and Route Messages from the CCFs.

Reference back to Figures 4 and 5 will prove useful in obtaining an overview of the software system. The highly repetitious nature of this system suggests the use of efficient modularized subroutines throughout. This will eliminate repetition of similar functions, such as similar I/O operations, and hence decrease core size. The frequently used subroutines will be core resident.

Conclusion

With the proliferation of telecommunications in the usage of computers there will exist an increasing need for sophisticated configurations and external configurations such as the Communication Interface System. The CIS will relieve the Central Computers from inefficient processing of error and validity checking, polling and other non-computational tasks. A hardwired multiplexor, such as the IBM 2701 can support four asynchronous or two synchronous lines. This alone costs $38,000 and going to a third synchronous line requires the more expensive IBM 2703.[8] As the number of lines increase, a small computer is increasingly justified on financial grounds alone.

It has been shown that a small computer can handle the communication needs of a larger system. The hardware has been specified and shown to exist through a short survey of the market. The configuration will function, as outlined, with the specified and projected load factors. Software will be mostly user-written, but is straight-forward.

In retrospect, this system has a workable design and a practical application. The Chase Manhattan Bank's System/70 Extended Computing System, designed by Mark Computing Systems and installed at Chases' UniCard Division is currently operational.[9] Other systems, such as the Communication Interface System, will become increasingly evident in the future as communication needs expand.

APPENDIX I

Figures showing actual loading of reservation lines are not available. From actual experience it takes an average agent six minutes to complete a normal transaction. Two queries are first sent to the CCF and two answers received before an available flight is found. Then the appropriate data such as: Name, telephone, ticketing, etc. is entered and an acknowledgement received. Since a block will be 96 characters even if it contains less data, and no normal single transmission should exceed 96 characters, an agent will average one block or 96 characters/minute.

APPENDIX II

MACHINE EVALUATION

MACHINE EVALUATION

The suitability of various small computers on the commercial market will be analyzed as the concluding section of this thesis. Their specifications will be compared against the requirements that have now been developed. For the purpose of analysis the following machines have been selected for study: Digital Equipment Corporation PDP-9, Honeywell DDP-516 and the Xerox Data Systems Sigma 2. The machines will be compared in chart form in the following areas, with a discussion following each:

1. Memory
2. CPU Features
3. Arithmetic Operations
4. I/O Capability
5. Software
6. Miscellaneous features
7. Overall choice.

MEMORY

The XDS-2 surpasses the DDP-516 and PDP-9 in terms of speed of basic memory cycle time. It also has greater expansion capability and has a standard parity check feature. The PDP-9 does have two more bits in word length which could conceivably serve a useful purpose. Otherwise the units are effectively the same in our application.

CHOICE: Sigma-2

REASON: Although core size has been determined at 32K, the expansion capability given by the PDP-9 could conceivably be useful, in addition to a 9.1% faster memory cycle time than its closer competitor are the reasons. Since the DDP-516 and PDP-9 have an optional parity check, and all machines have an optional memory protect feature, there is no apparent difference here. The 16-bit word size is deemed adequate for our purpose and the two extra bits on the PDP-9 are not useful.

Feature	PDP	DDP	XDS
Memory cycle time (u sec)	1	.96	.9
Memory word length	18	16	16
Minimum memory size	8K	4K	8K
Memory increment size	8K	4K	4K
Maximum memory size	32K	32K	65K
Parity check	opt.	opt.	std.
Memory protect	opt.	opt.	opt.

CENTRAL PROCESSOR

The PDP-9 eliminates itself in this area due to the time required for a direct (single) access (100% more than the XDS, Sigma 2 or the DDP-516) or an indirect (double) access (a 50% increase over the others). The XDS machine has two more registers than the Honeywell and is faster by a small margin. The DDP-516 has multi-level indirect addressing versus the single-level addressability of the Sigma 2.

CHOICE: DDP-516

REASON: The Sigma 2 is only slightly superior in terms of speed and register availability. Multi-level addressing available on the DDP-516 will prove useful for structuring of queues and pointers to records.

Feature	PDP	DDP	XDS
Instruction word length	18	16/32	16
Accumulators or GP Reg.	2	2	2
Hardware Reg.	2	5	6
Index Reg.	7	1	2
Bits for Op. Code	4	5	4
Bits for Address Mode	1	2	4
Addressing Modes	2	4	16
Bits for address	13	9/14	8
Directly addressable core	8,192	1,024	1,024
Time for above (u sec)	2.0	.96	.9
Indirectly addressable core	32K	32K	65K
Time for above (u sec)	3.0	1.92	1.8

ARITHMETIC OPERATIONS

All three machines under consideration are almost equal in these aspects. Since the application of the CIS is primarily I/O handling and not scientific, floating-point hardware is not necessary. The DDP-516 does have a basic full word store time approximately 4% faster than the next machine.

CHOICE: DDP-516

REASON: All other factors being equal, the Honeywell machine is faster in most applications.

Feature	PDP	DDP	XDS
Store time for full word (u sec)	2.0	1.92	2.2
Add time for full word (u sec)	2.0	1.92	2.2
Fixed-point Hardware mult./div.	opt.	opt.	opt.
Fixed-point Mult. (u sec) hardware	3.-11	5.28	10.3
Fixed-point Div. (u sec) hardware	3.-12	10	10.8
Fixed-point Mult. (u sec) software	281 max.	154.6	103.
Fixed-point Div. (u sec) software	352 max.	220.8	297.

INPUT/OUTPUT CAPABILITY

The primary orientation of the Communication Interface System is an I/O handling for the larger Central Computing Facilities. Therefore speed and capability in this section is critical. Since only one 6-bit data character will be stored per word the 8-bit Data path of the XDS Sigma 2 is not detrimental. The Sigma 2 has a DMA transfer rate five times slower than either of the other two machines. However, it still comfortably exceeds the 80K Hz requirements. The PDP-9 has an interrupt response time more than twice that of the DDP-516. The number of external interrupts was determined to be 48 which the Honeywell meets exactly and the PDP-9 surpasses by 208 interrupts.

CHOICE: PDP-9

REASON: The PDP-9 is most desirable because the CIS will handle a great many interrupts and provides a margin for expanding the number of interrupts.

Feature	PDP	DDP	XDS
Direct Path Width (bits)	18	16	8
DMA Channel	std.	opt.	std.
Maximum DMA Transfer Rate (KHx)	1000	1000	200
External Priority interrupt levels	1	2	2
Maximum number of external interrupts	256	48	132
Response time (u sec) including time to save registers of interrupted program and initiate new program	4.	9.6	6.

MISCELLANEOUS

All three machines either have the features noted below or provide them as an option. The real-time clock, restart capability and power failure protect features are all necessary.

CHOICE: DDP-516, PDP-9 or Sigma 2

REASON: There is no apparent difference in any of the machines to perform the required tasks.

Feature	PDP	DDP	XDS
Power Failure Protect	opt.	std.	opt.
Automatic restart after power failure	opt.	opt.	opt.
Real-time clock	std.	opt.	opt.

SOFTWARE

Due to the specialized needs of the CIS, the real-time monitor system will be written specifically for the application, and the manufacturer's executive will not be used. Therefore the sole need is for a viable assembler and a disc operating system, if a suitable one exists. All coding is to be in assembly language, eliminating the need for compilers or conversational compilers.

CHOICE: DDP-516, PDP-9 or Sigma 2

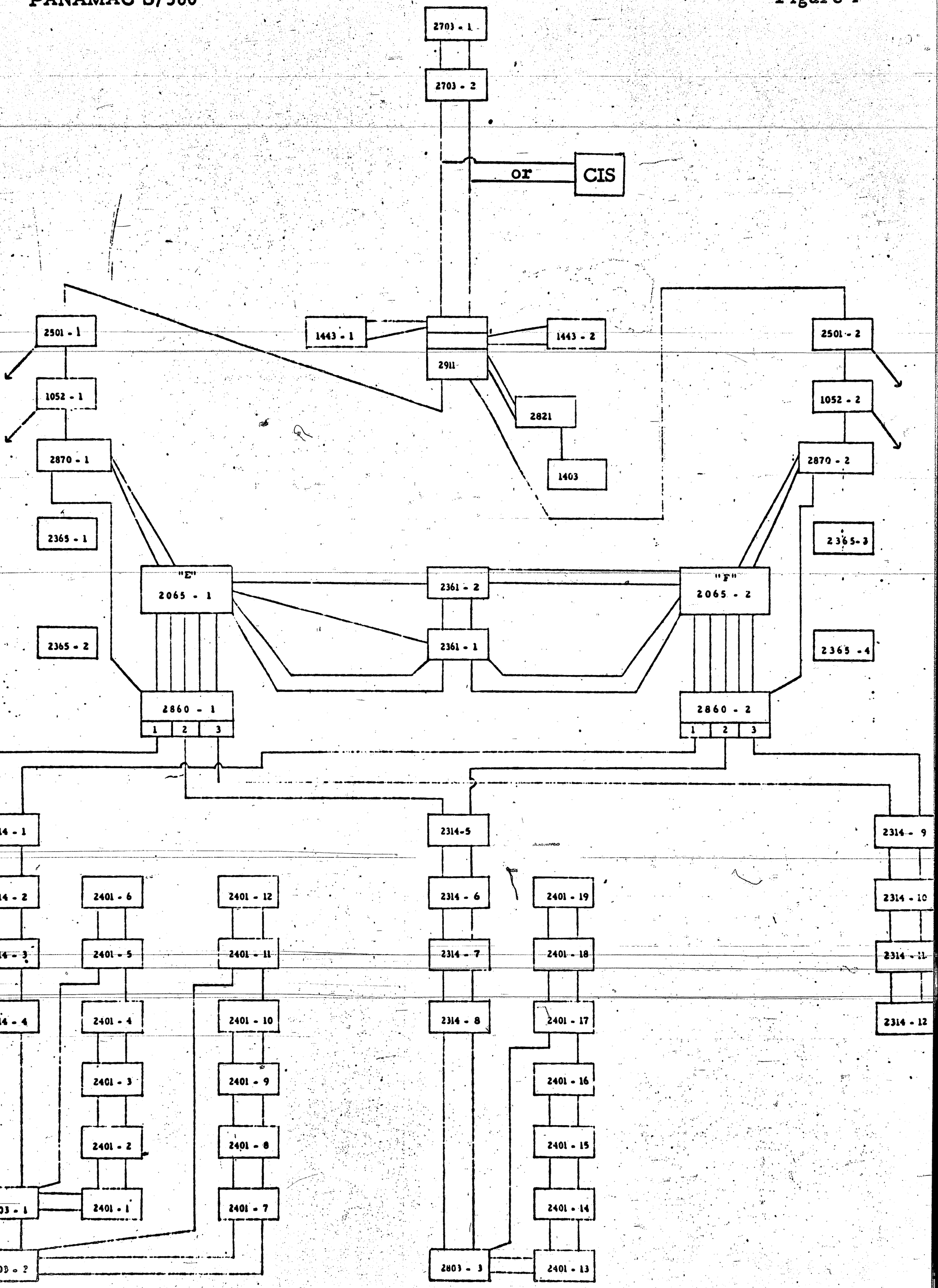
REASON: All applicable features appear equal in software supplied. Assuming that the quality of the respective packages are acceptable, any of the three will be satisfactory. However, this assumption would bear further investigation. Often manufacturer supplied software is found lacking. User written software such as LEDAP, developed at Lehigh University, are often far superior and more suitable.

Feature	PDP	DDP	XDS
Assembler (1 or 2 pass)	2	both	1
Relocatable Assembler	yes	yes	yes
Minimum core for above	8K	n/a	8K
Macro capability	yes	no	yes
Compilers available	Fortran IV	Fortran IV	Fortran IV BASIC
Conversational Compilers	FOCAL	Fortran IV BASIC	none
Real-time executive	yes	yes	yes
Disc Operating System	yes	yes	yes

Machine Evaluation Conclusion

All of the units reviewed are sufficient for the requirements defined for the CIS. None of the units is disqualified for a severe short-coming, although in various areas one may be preferable. Many other small or medium size computers such as the Supernova and Nova Computer have been designed and integrated into a front-end environment.

APPENDIX III - FIGURES


2703 - 1
2703 - 2
or
CIS
1443 - 1
1443 - 2
2911
2821
1403
2501 - 1
1052 - 1
2870 - 1
2369 - 1
2365 - 2
"E"
2065 - 1
2860 - 1
1 2 3
2501 - 2
1052 - 2
2870 - 2
2365 - 3
2365 - 4
"F"
2065 - 2
2860 - 2
1 2 3
2361 - 2
2361 - 1
2314 - 1
2314 - 2
2314 - 3
2314 - 4
2401 - 6
2401 - 5
2401 - 4
2401 - 3
2401 - 2
2401 - 1
2401 - 12
2401 - 11
2401 - 10
2401 - 9
2401 - 8
2401 - 7
2803 - 1
2803 - 2
2314-5
2314 - 6
2314 - 7
2314 - 8
2401 - 19
2401 - 18
2401 - 17
2401 - 16
2401 - 15
2401 - 14
2803 - 3
2401 - 13
2314 - 9
2314 - 10
2314 - 11
2314 - 12

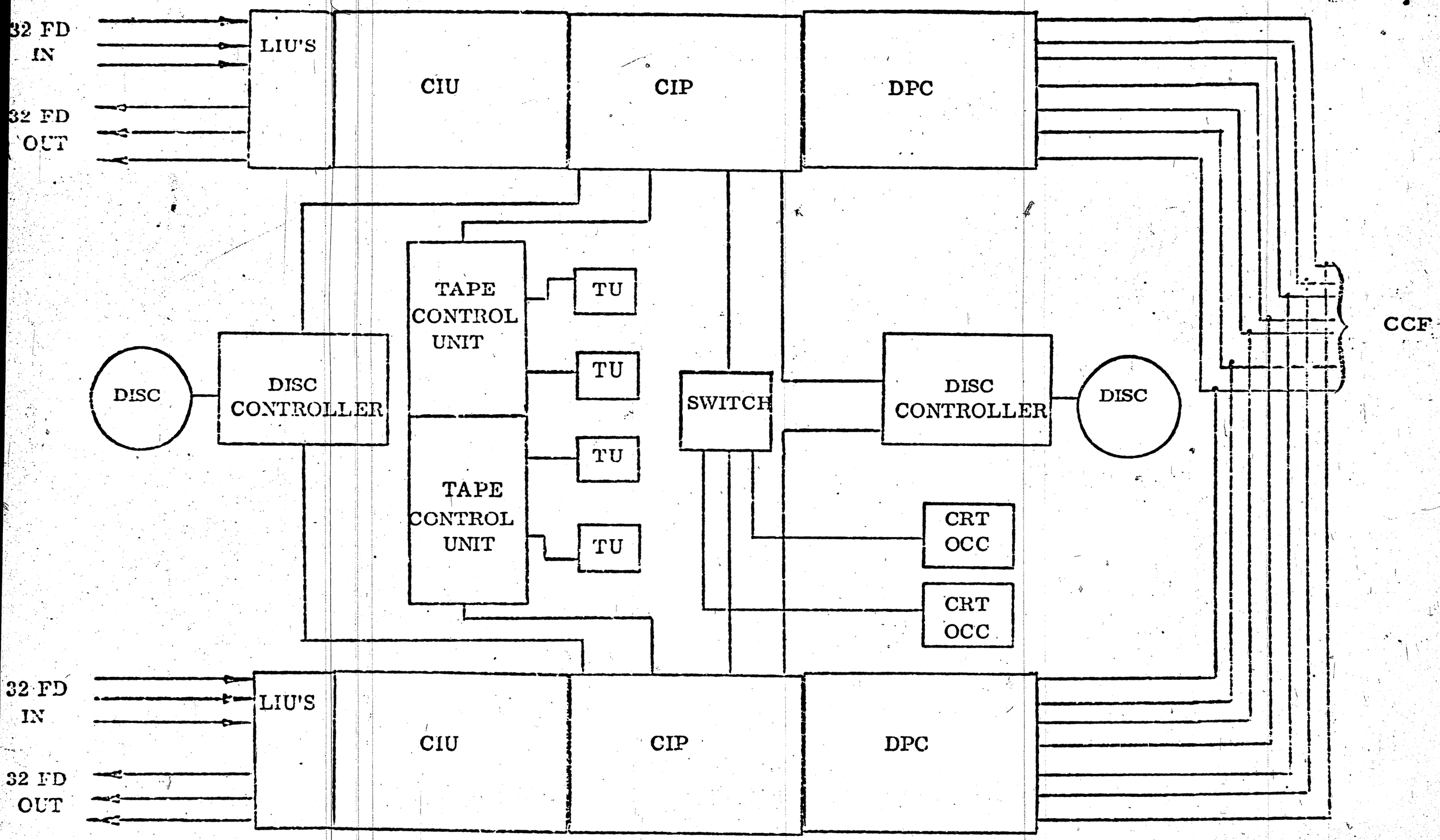


Figure 2
Communication Interface System

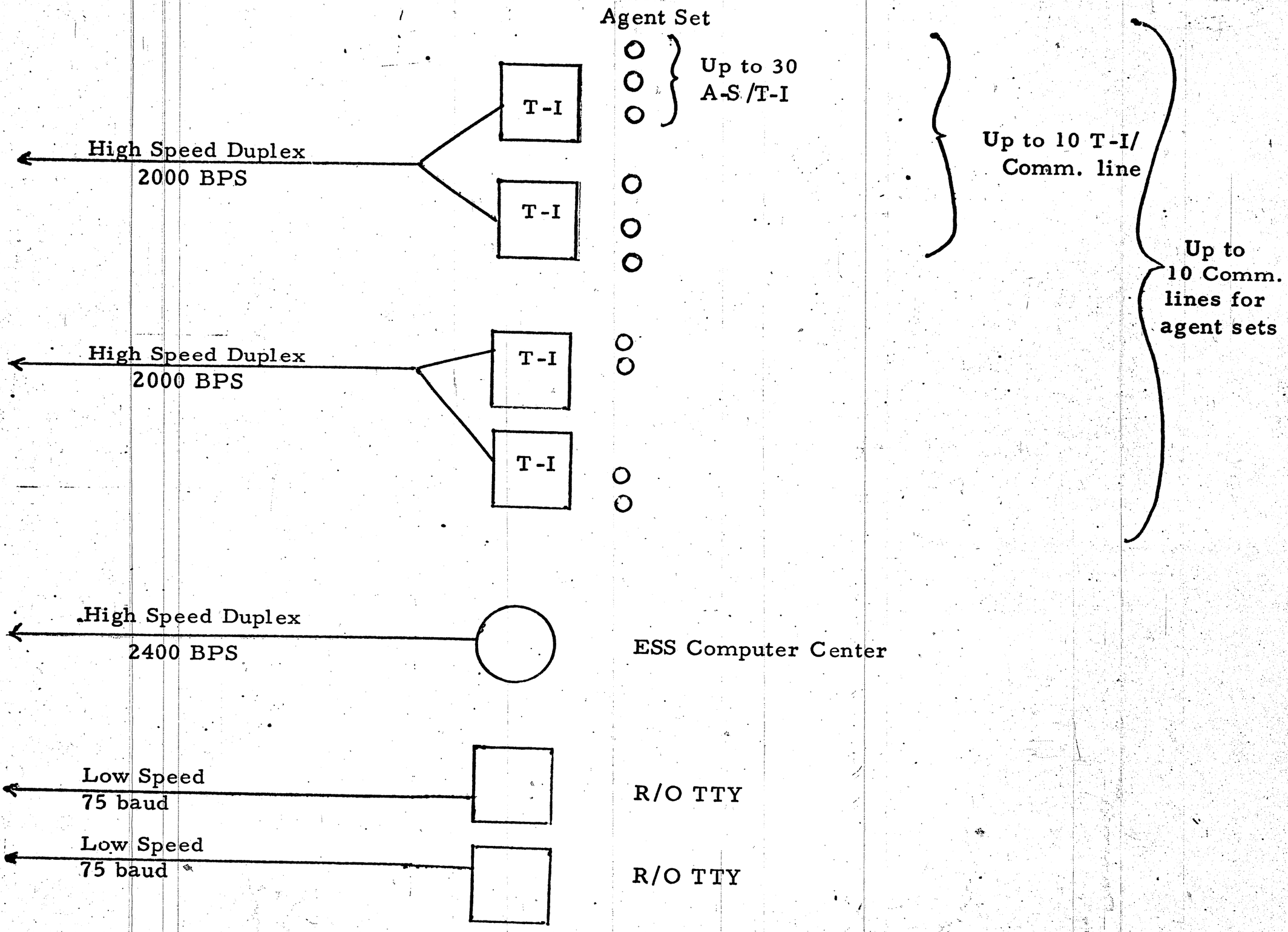
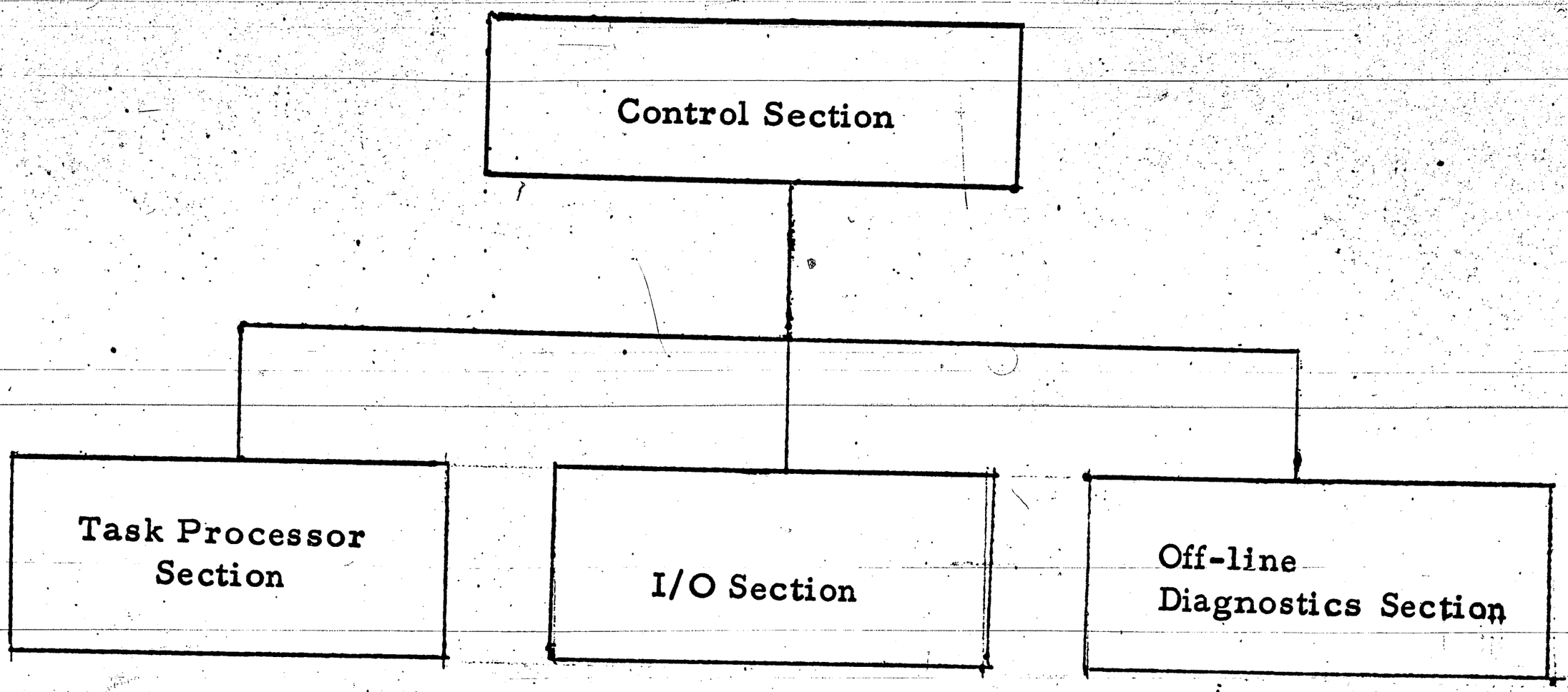


Figure 3
Communication Line Configuration

Major Program Sections



CONTROL SECTION STRUCTURE

I. Utility Functions Module:

a. Systems program loader
b. Core Dump Routine
c. Core Flush Routine

II. Initialization Module:

a. CIS OCC
b. Data Logging
c. Initial Entries
d. Buffer Chain Generation
e. Network Input
f. Recovery File

III. Common Functions Module:

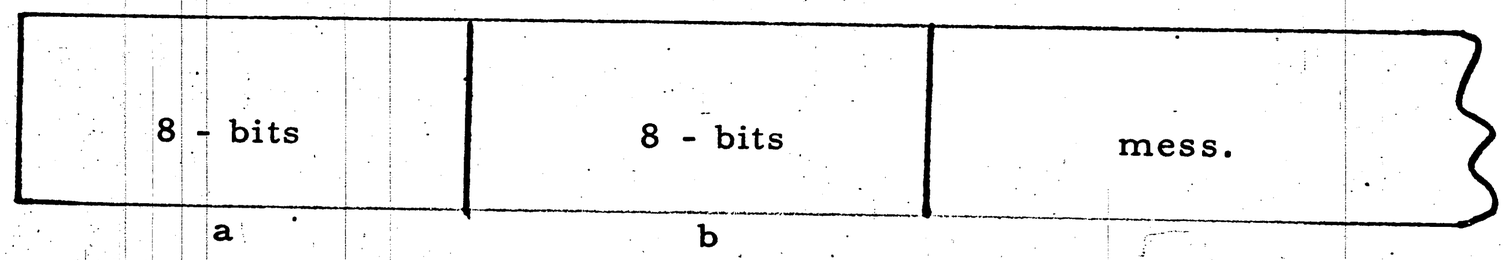
a. Dynamic Core Storage Allocation
b. Magnetic Tape Functions
c. CIS OCC I/O
d. Data Logging

IV. Periodic Functions Module:

a. Basic Cycle Reset
b. Task Timing Update
c. Task Controller

V. Non-Periodic Executive Module:

a. Task Controller
b. Message Queue Supervision



a. Line address (mess. org. or des.)
not all ones or zeroes

b. CPU # or other useful data

Figure 6
Message Header Identification

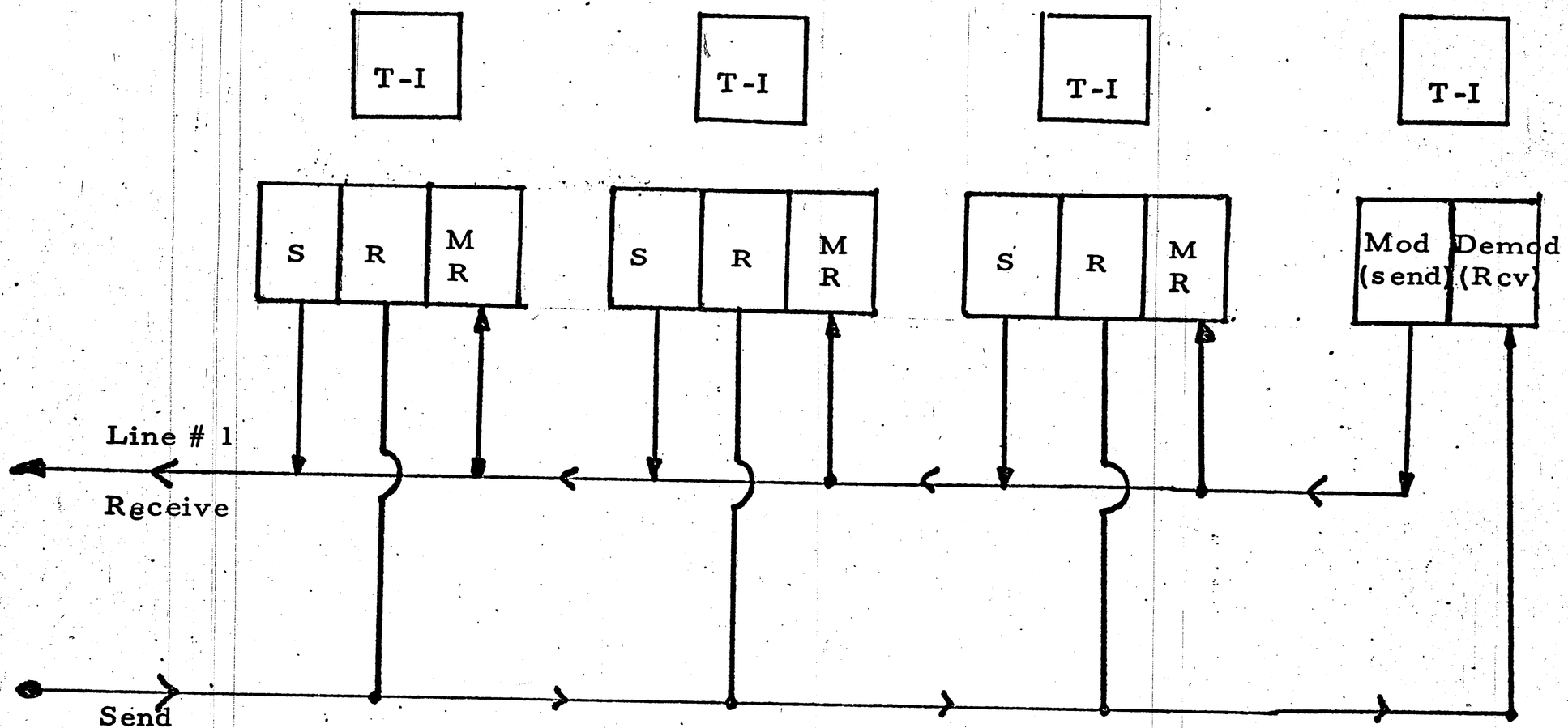
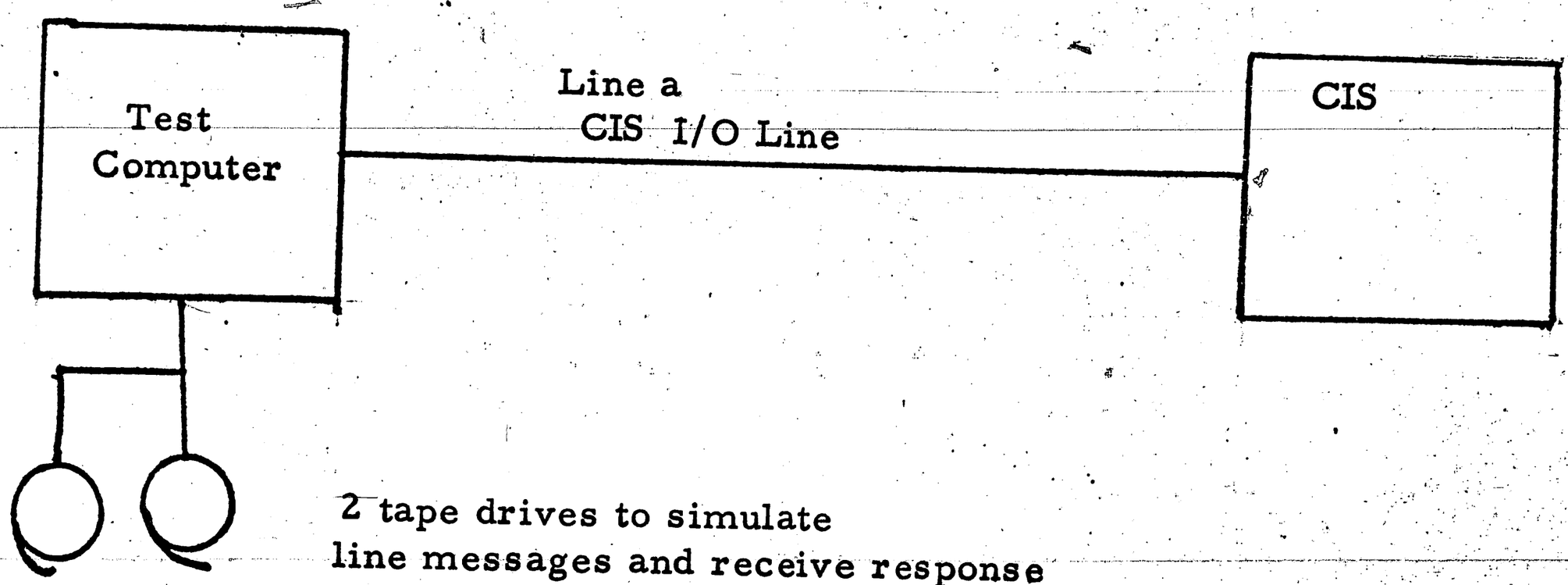


Figure 7
Terminal Interchange Configuration

Figure 8

Simulation Test Configuration



APPENDIX IV - FOOTNOTES

FOOTNOTES

1. The information contained in this paper is not intended to represent the plans of Pan American World Airways, Inc. Rather, it is drawn in part from studies done by Pan Am and the General Dynamics Corporation and contains changes and this author's ideas.

2. The teletypes will receive system-wide status reports such as polling of Terminal Interchanges.

3. Four channels are based upon the DDP-516 which requires a DMA channel for each set of eight communication lines.

4. Parallel transmission is based on an engineering modification to the Computer.

5. Through use of assembly language, generation of excess coding, inherent in most high-level languages, will be avoided.

6. Datamation, March 1969. "Minicomputers for Real-Time Applications" by D.J. Theis and L.C. Hobbs.

7. IPARS is IBM's International Passenger Airline Reservation System.

8. Computerworld, July 1, 1970. "Mini Front-Ends Provide Alternative to Multiplexor" by Jim Pearlman.

9. ibid.

BIBLIOGRAPHY

Mini -Computers for Real-Time Applications,
by D. J Theis and L. C. Hobbs, in "Datamation",
March 1969, Thompson Publications, Inc., Greenwich, Connecticut.

Mini Front-Ends Provide Alternative to Multiplexer,
by J. M. Pearlman, in "Computerworld", July 1, 1970, Newton,
Massachusets.

Technical Proposal for Panamac Communication Interface System
Definition, by General Dynamics Corporation, January, 1969,
San Diego, California.

Technical Proposal for Pan American Airways Interim Communication
Interface System, by General Dynamics Corporation, June 1969,
San Diego, California.

Pan American Airways Communication Interface System Design
Document, by General Dynamics Corporation, May 1969,
San Diego, California.

Communication Interface System Computer Specifications, by
General Dynamics Corporation, San Diego, California.

Interim Communication Interface System, Computer Program
Specifications, by General Dynamics Corporation, June 1969,
San Diego, California.

Interim Communication Interface Hardware Specifications, by
General Dynamics Corporation, August 1969, San Diego, California.

Honeywell H-316 General Purpose Digital Computer Programmers
Reference Manual, Honeywell, Minneapolis, Minnesota.

u -Comp DDP-516 General Purpose Digital Computer
Programmers Reference Manual,
Honeywell, Minneapolis, Minnesota.

DDP-516 Interface Manual,
Honeywell, Minneapolis, Minnesota.

DDP - 516 Users Guide,
Honeywell, Minneapolis, Minnesota.

DDP - 516 Installation Manual,
Honeywell, Minneapolis, Minnesota.

VITA

Robert J. Schwartz, Jr. was born on August 29, 1946 in Washington, D. C. His father, Robert J. Schwartz PhD, is an Economist and his mother, Beatrice Sheiner Schwartz, is a Free-Lance Commercial Artist.

Schwartz graduated Great Neck North Senior High School in Great Neck, New York in June 1964. He entered Bucknell University, Lewisburg, Pennsylvania in September, 1964 and graduated in June 1968 with the degree of Bachelor of Science in Mathematics.

He attended Lehigh University Graduate School, Bethlehem, Pennsylvania from July 1968 until August 1969. Upon entrance until July 1969 he held a graduate assistantship at the Lehigh University Computing Center. During the month of July, Schwartz was a Research Assistant at the Lehigh's Center for the Information Sciences.

From August 1969 until May 1970 he was a Computer Systems Analyst and Marketing Supervisor, Computer Time Sales with Pan American World Airways in New York City. Since May 1970 he has been Director - Far East Operations for Harman Kardon in Plainview, New York and Tokyo, Japan.