OUTLINE OF FRONT END REQUIREMENTS FOR A CDC-6400 COMPUTER

George Yatrou

TECHNICAL REPORT

in

The Faculty

of

Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering at
Sir George Williams University
Montreal, Canada

May, 1974

OUTLINE OF FRONT END REQUIREMENTS FOR A CDC 6400 COMPUTER

GEORGE YATROU

ABSTRACT

This paper outlines the requirements for a mini-computer based front-end used to control the telecommunications lines connecting user terminals to the CDC-6400 computer system of Sir George Williams University. The paper comprises two basic parts.

The first part expresses the author's view regarding the state of the art in computer communications. This is done by outlining how developments in each component of this industry contribute to the growth of the industry as a whole, and finally result to benefit for users of such systems.

The second part proposes that a DEC PDP11 would be a most suitable alternative for front-ending the SGWU system. To this end, the basic principles of operation and the merits of that system are outlined. Finally a method of connecting the PDP11 to the CDC system is suggested.

ACKNOWLEDGEMENTS

The author wishes to thank Prof. G. Martin for his guidance and support throughout the development of this work. Thanks are also extended to Messers L. Shallit of Digital Equipment Corporation, Jacques Thibeault of the University of Montreal, and Pierre Aubè of Allan Crawford Associates for providing information and advice.

CONTENTS

I. INTRODUCTION

A few years ago only a few computer installations provided service to large numbers of users. The state of the art in computer, telecommunications and terminal equipment made direct computer access economically unfeasible. However, such direct access and use of computer complexes has now overcome many obstacles, mainly because of technological development in all major sectors of the user-to-computer industry.

The user's position has improved because of technological advancements in the telecommunications industry. Progress in this industry has made possible speeds and data throughput to satisfy most subscribers. In terms of cost performance, advanced technology has reduced the cost of purchase, operation and maintenance of common carrier equipment and facilities. Development of improved or new methods of transmission, made it possible for common carriers to save on the required amount of equipment and facilities. Advents such as digital networks, time division multiplexing of large numbers of users into high speed circuits form the basic backbone of telecommunications engineering.

Developments in this area have produced the Dataroute, Infodat or their USA equivalents. These developments have changed the tariffs and in general the whole subscription rate structure of Telecommunications companies. The net result is that users can now transfer data, at a fraction of their cost as of about five years ago.

Improved line disciplines to obtain a greater and more economical choice of terminal or peripheral equipment, contribute tangible benefits for the end user. There is a strong trend towards the development of remote batch terminals. Since remote batch terminal complexity may vary from a simple teletype machine to a substantially large computer system, and countless options in between, the user can choose the entry system which suits his needs best. In terms of economics, the batch terminal user, besides doing away with having to have his own computer system, realizes savings from ever-decreasing purchase costs for terminal equipment. Cost reductions in this area are less than they could be, because of lack of standardization of requirements, and consequent difficulty to produce it in mass for greater cost reductions. Instead, many users still use custom made terminals developed for one particular use only. It is hoped that the industry shall expand so that the most common terminals will be mass-produced, and avail to the user the benefit of the economics of mass-production.

Since the primary objective of Computer Systems is problem solving and handling of data bases for their users, many systems owners and operators has relieved the central computer from the burden of interfacing with each and every user. Therefore, many systems are equipped with special units for communications processing. Depending on the extent of responsibility of these processors the user-to-computer interface may vary from the 270X group to complex front end systems such as the IBM-3705, the Interdata type 55 or others using powerful computers such as DEC's PDP-11. The Sir George Williams University computer, CDC-6400 system equipped with the Kronos Cyber 70 operating system, uses the CDC supplied Peripheral Processing Units. It is intended here to develop the guide lines for a free standing front-end processor, which will control the present communications system and anticipated expansion more flexibly and economically.

II. DESCRIPTION OF THE PRESENT SYSTEM

The CDC-6400 computer system presently installed in the SGWU computer centre interfaces and controls I/O devices, by means of a system comprising ten peripheral and control processors. These processors are stand-alone programmable units, used to control the mainframe peripheral devices as well as the system I/O devices. All peripheral processor units (PPU) communicate with external equipment and each other on twelve independent I/O channels. Data flows between the PPU and the external I/O device in blocks of words, whereby each word is conveyed to the external device via the A register of the PPU connected to the device.

Flow of data between the processor and the central memory is also in blocks of words, but in this case 60-bit central memory words are used as the basic element instead of 12-bit PPU words. Each five PPU 12-bit words are assembled into one 60-bit word, before its transfer to central memory, and conversely each 60-bit word from central memory is disassembled into five 12-bit words prior its transfer to the PPU memory. To enable all ten PPUs to transfer data to central memory a multiplexing system, comprising a so called 10-position barrel is used. This allows all ten PPUs to process their programs on a time shared basis. As a result each PPU has access to the CPU once every major cycle, which is the time

required to go through the ten program steps to go around the barrel. Since the major cycle is about 1000 nsec the maximum transfer rate, between the PPU and the CPU is approximately one million words per second.

To direct activity associated with data transfers between the PPU and the external I/O devices, each PPU is equipped with 12 instructions. These instructions are used to determine whether a channel or external device is available and ready to transfer data. The equipment status is of course ascertained by means of the device's function codes. The following program shows the steps typically involved to transfer data between the PPU and any I/O device connected to one of the channels.

SP-1 CDC-6600 I/O DATA TRANSFER PROGRAM

INSTRUCTION	OP. CODE NO.	MNEMONIC	EXPLANATION
1.	65	IJM md	Jump to (function code) m if channel d is inactive
2.	77	FNC md	Perform a function m on channel d, for example check its status
3.	74	ACN d	Activate channel d
4.	70	IAN d	Input to A from channel d, information about the status of the desired I/O device
5.	75	DCN d	Disconnect channel d, precaution against PPU hanging up.
6.	77	FNC md	Function m on channel d, to prepare the desired device for data transfer.
7.	30	LDD(d)	Load A with (d), where (d) = number of words to be transferred.
8.	74	ACN d	Activate channel d. Set channel active flag and prepare for impeding data transfer.
9.1	71	IAM md	Input (A) words from channel d to m.
9.2.1	73	OAM md	Output (A) words from m to channel d
9.2.2	66	FJM md	Jump to m if channel d is full
10	75	DCN d	Disconnect channel d and make it inactive

Remote terminal users access the system via the 6671-A Data Set Controller, which can control up to sixteen modems operating in a full or half duplex mode. The DSC control disassembles and transmits, or receives and assembles 8-bit characters for data transfer between the desired terminal and the DSC output, or input buffer registers respectively. As a working area, the DSC uses a 64-word, 28-bits-per-word core memory. Fig. 1 shows how this memory is organized.

DSC Memory Location	00	01	10	11
0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111	Data Storage for Data channel	Output b information (for b disassembly)	Input information (for assembly)	Memory Parity character for each modem.

Fig. 1. DSC Memory Partitioning

The activity which controls and transfers data between the PPU and the DSC is loosely summarized in the following steps.

1. The PPU issues to the DSC a function code by which it selects the desired terminal and assigns the required operating mode. This function code contains information to control the selected equipment in the required operating mode and to also control the actual data transfers.

2. In reply the DSC issues an Inactive indication and enables the requested mode of operation or sets flags to indicate malfunctioning or other abnormality.

3. When the PPU receives the DSC ready indication, it activates the data channel and transfers and stores a block of data in the DSC output buffer register (OBR). The DSC then readies the terminal, and transfers the stored data to the output disassembly register (ODR) for transmission to the line.

4. Data received from the line is assembled in the input assembly registers (IAR), and then it is transferred to the PPU via the input buffer register (IBR).

Comparison shows that this process and the sample program could be made identical. The only difference is in that one reflects the operation from the DSC's viewpoint and the other from the PPU's viewpoint. Since the terminal equipment is connected to the system via modems with standard EIA-RS-232 interface, there are no restrictions on the type of terminals used.

110 baud lines interface the system via the type CDC-6676-B multiplexer, which controls up to 64 modems. The transfer of data between the data channel and this multiplexer is basically the same as for the 6671 data set controller. The differences between the two are:

1. speed of operation,
2. the 6676-B multiplexer operates asynchronously, as opposed to the 6671-A
3. the 6676-B can control a maximum of 64 modems instead of 16.

The CDC-6400 system in the SGWU computer centre building, comprises a type 6400 central processor, 98K words central memory (60 bits/word) and a group of 10 PPU's. The PPU's are interconnected

1. with the computer via a 10-port multiplexer and

2. with the peripheral equipment, the I/O equipment and each other with 12 type 6000 data channels.

III. SYSTEM CONSIDERATIONS

Telecommunications

Development and growth of the telecommunications industry has enabled its users to realize cost reductions of about 50%, because manufacturers have improved their manufacturing methods and techniques. As an example, table 1 gives purchase cost changes for modems in the last five years.(1)*

TABLE 1. PURCHASE COST CHANGES FOR MODEMS SINCE 1968.

Type of Modem	Speed in Baud	Present Cost - $	1968 Cost - $	Reduction %
103	300	200	350	45
202D	1800	325	650	50
201/B	2400	980	2500	60
-	4800	4000	7500	45

As a second example, limited distance data sets have made it possible to replace expensive 4800 or 9600 baud modems by units costing under $1,000. With these sets error rates less than 1×10^{-5} can be achieved for transmission of 4800 baud over five miles of 24 AWG cable pair conductors.

*Represents reference number

The other source of economic benefit in telecommunication comes from the telephone company and its competitors. Developments in common carrier facilities, methods of transmission, and in facility organization have resulted to benefit for both the user and the communications companies. Tariff reductions are enhanced by the creation of small specialized companies which compete only for portions of the market, and force the major common carriers to provide similar services or improve their existing. Switched data networks such as that offered by DATRAN in the USA and digital networks such as Dataroute or Infodat have resulted in tariff reductions of about 50%. It has turned out, that the major recipients of the benefits from digital networks, are users of low speed equipment. This, along with the tremendous growth of the low speed terminal (teletype and CRT) industry, are setting a trend in that direction.

Interconnection and interfacing of data terminal equipment to telecommunications lines is made by hard wired modems, coupled digital to analog converters or by direct connection of teletype machines to telegraph grade lines. The choice will depend on the speed of the terminal equipment and its distance from the point of destination.

The communications lines may be leased or dialed over the telecommunications company's switched network. The criteria for choosing leased facilities are large volume of

data for transmission, long distance tolls, and data speed limitations.

Since the bulk of terminal equipment used on the SGWU system is low speed, and located in the Montreal area, there is no reason for leased facilities. The present arrangement, using 103 type modems or acoustical couplers over Bell Telephone dial-up lines will suffice. Higher speed modems, such as 2400 baud or 4800 baud will also operate without difficulty over the same facilities. However in the case of synchronous modems the choice of leased or dialed facilities should be investigated for each application.

Communications systems also combine, or concentrate many low speed inputs into one high speed output stream, by using frequency division or time division multiplexers. The modern trend is towards time division multiplexing, and the state of the art in this field calls for very sophisticated software operated equipment.(2)

Terminal Equipment

The I/O equipment of the SGWU computer system comprises, one 1200 card per minute reader, two line printers (1200 and 1000 lpm), one Tektronix 4010 graphic display, 40 time sharing ports (10 and 30 cps) consisting mainly of TWX and Vucom terminals, and four remote batch ports connected to two 200 user terminals (one at 4800 baud and one at 2000 baud), one

PDP/8 computer (simulating a 200 user terminal) at 4800 baud, and one 4800 baud line to Loyola.

It is believed that the system in the next three or four years will triple. The types of terminal used is not expected to change, but economics will be the predominant factor in the choice. The basic subdivision in two types with and without hard copy is expected to remain. Technological improvement in either category and associated benefits are of course expected, since the industry of terminal equipment[3] manufacture is growing rapidly.

Line Interface Requirements

Communications lines will interface the I/O channel of the front end by some unit which can provide the channel with the elementary control signals such as:

1. Status character: to indicate request for service or if the device is in the idle, receiving or transmitting mode. Other information such as code used, speed of I/O can also be incorporated in the status character.

2. Line interface. This is usually accomplished by modems operating in accordance with EIA RS-232 standard recommendation. It is necessary here to provide speed compatibility with the remote equipment.

3. Buffering which enables block transfers and enhances speed and efficiency of data transfers. In addition buffering reduces the probability to lose characters. It also creates an easier environment for the programmer, since there is no need to request service for every character received or transmitted.

4. Synchronization. The line interface can be equipped to insert or delete start-stop bits or synchronization characters automatically. Provision must be made to change by software the sync character, or the length and the number of start-stop bits as required by the code of the device.

5. Clocks for timing data transfers between the modem and the interface. The speed of the clock should be program selectable.

6. Facility for incorporating hardware for error correcting, such as character echoeing for lines using 103 types of modems, reverse channel acknowledgement for lines using 202 types of modems and full-duplex transmission for connection to four-wire modems.

7. Facility for testing communications lines and associated equipment.

Communications Processor

The objective of this project is to suggest a minicomputer system to front-end the CDC-6400 system by connecting it to one or both of the presently used PPU's. Since the front end will now handle most of the PPU functions one PPU will suffice to transfer data between the front end and the host.

To accomplish this the following problems must be solved: (a) develop a method of interconnecting the proposed processor to the data channel and associated PPU, (b) select a free-standing communications processor, preferably a minicomputer, with appropriate hardware and software to provide a user transparent system and (c) provide means of interconnecting the telecommunications lines to the system without the need to remove any of the presently used communications devices.

The communications processor industry has made available for users a multitude of products (4) ranging from a simple multiplexer directly connected to the computer, to IBM-270x line controllers, to sophisticated computer systems controlling communications networks or vast switching systems in telephone or data exchanges. The problems in selecting a communications processor are of two types. One type concerns the choice of a system mainly designed for the proposed application, or a general purpose system which can be made to fulfill

the requirements; the other type concerns the size of the processor, in terms of capability, storage, software, and peripheral equipment.

The choice between a general purpose minicomputer and a specially designed communications controller depends on deciding which system can be operated, maintained, modified and expanded economically and easily. It is believed here that a powerful general purpose minicomputer can be used to design a superior front end system. The following paragraphs are expected to lead to that conclusion.

Specially designed systems are usually hard-wire or low-level program operated. Alteration and maintenance of such systems requires skilled personnel to perform tedious and difficult operations. Such personnel and changes are generally expensive. In the long run, unavoidably, the system shall require such expensive interventions, and the balance will be tipped towards the more flexible general purpose computer, and ordinary programmers. It is accepted that the efficiency of a front-end system based on a general purpose minicomputer will be lower. Using a hard-wired instruction specially designed and installed for one specific function will be more effective than adapting and using one or more instructions in the general purpose minicomputer. The obvious question at this point is, how important is efficiency? For this particular installation it will turn

out that the general purpose minicomputer front-end will have sufficient capacity. With this in mind, it is decided to propose a general purpose minicomputer front end, where efficiency, for whatever it is worth, and initial economy are traded off for flexibility, and the basis to save money and effort in the long run.

In the previous paragraph it was decided to choose flexibility instead of initial savings and efficiency. The specific areas of the minicomputer system which will provide this flexibility are: (a) hardware and software modularity to allow expansion and modification, (b) level of software available which decides the ease of modification. If a system can have its memory increased economically, say $5,000/4K installed, one could justify using high-level programming for the system. Therefore, when the need arises a module of memory and a software module, developed by the system's programmers or by a software house, could prove very useful. Today it is possible to equip 16 384 words of memory on a single printed circuit board, and one can buy an off-the-shelf communications program for a PDP-11 or Xerox Data Systems from Informatics Systems. This type of ready availability can only continue to grow and make it less expensive for users to operate and maintain their systems.

As it was mentioned above, special application systems are hard to justify because their advantage in efficiency and low initial cost is offset by the flexibility of the general purpose machine. Highly powerful machines such as the PDP-11 or the system 55 of Interdata provide the best means of solving computer communications problems. In addition, dealing with companies like Digital Equipment Corp., Interdata, or Data General practically eliminates the chance of the supplier discontinuing operations and disrupting service and support for the supplied equipment. Those companies have a very large product and user base, and therefore they are unlikely to discontinue operations without guarantee for service of the equipment they have sold.

IV. FRONT END PROTOCOL

General

The proposed front-end will be required to:

1. Transmit and receive all data passing through the system.
2. Pro-and-post-process the transferred data.
3. Control the communications network and
4. Provide a fail soft capability for the communications portion of the system in case of host failure.

Data Transfers

During data transfers from the lines to the computer system the front-end will:

1. Assemble received data from its serial-by-bit format into words or bytes as required.
2. Strip start-stop bits or synchronization characters.
3. Transfer assembled characters from the input line buffer directly into central memory or into mass storage.

4. Pre-process stored received data and forward on to the host computer in multiples of 12-bit words.

During data transfers from the host-computer to the lines the front-end will:

1. Receive outgoing data from the PPU in multiples of 12-bit words for disassembly into bytes or front-end words and post-processing.

2. Transfer post-processed data to the line output buffers.

3. Disassemble characters into serial-by-bit data and transmit to the line.

4. Maintain synchronization for the communications equipment by adding start-stop bits or sync characters for asynchronous and synchronous transmission respectively.

Because the program controlling data transfers will be used much more often than other programs it is desireable to be:

1. re-entrant, to reduce program control requirements for each data transfer,

2. as quick as possible, to increase the system's

throughput. This can be accomplished by special instructions, microprogramming techniques, read-only memory storage for data transfer instructions, by using multiword buffers in the line-interface for group transfers, direct transfer to memory or mass storage for CPU time economy etc. The present system can transfer data at rates of about 0.25 Mbytes/sec and requires approximately 20 microseconds for establishing the connection to the required I/O device. The proposed system should at least match this throughput.

Line Control

The line control functions of the system are:

1. Originating calls by providing the line interface equipment with the proper function codes which can be used to control the modem via the EIA interface control leads.

2. Answering calls by monitoring and decoding the line interface status words which have been set-up by signals received from the modem over the EIA interface control leads.

3. Maintaining the connection during transmissions by maintaining "request to send" or monitoring incoming data and until an "end of transmission"

character is generated by the remote end or the host computer.

4. Recognizing line control characters and reacting accordingly. USAS standard X3.4 proposes a method of line control for lines using USASCII code. Similar procedures should be available for other codes.

5. Polling, auto-dialing and other communications network control procedures.

Data Processing

With the view to relieve the PPU load the front-end will perform data processing functions such as:

1. Code conversion. All received data should be translated from source to object code and transmitted data shall be translated from object to source. Code conversion should be programmable. One of the simplest methods for code conversion would be to equip the system with code tables, whereby characters can be translated by selecting the corresponding character from the required code table. The selection algorithms can be established by considering which codes will be equipped. USASCII, five level Baudot, BCD, and EBCDIC, are some of the most common codes and

must be equipped in this system.

2. Transmission error control features, such as parity checking or other coding schemes may be used to provide the system with the capability to recognize a transmission error, and if possible correct it. Some of the error control features which may be equipped with this system are (a) even or odd parity checking, (b) vertical and longitudinal redundancy character assembly and checking (c) coding techniques, such as cyclic or polynomial coding whereby errors are detected by logical operations on the received data. When an error is detected steps may be taken to correct it by (a) transmitting a message to the originator over the receive pair of the full duplex four-wire lines (b) transmitting a message over the reverse channel of communication lines using WE-202D modems, and (c) by simply feeding back the received data on lines using type 103 modems. For the sake of completeness the system must be equipped for a full complement of error control options as described above, with the exception perhaps of sophisticated coding techniques.

3. Queuing. Data inputs from communications lines shall be placed in a queue of priority for further processing by the system. The priority of each input will be derived from the users' identification which contains such information.

4. Editing and formating. The received or transmitted data will contain information pertaining to blocking and organization for presentation to the host computer or the user. The front-end shall extract this information and will organize the data in logical blocks such as programming statements etc. or physical blocks such as lines etc.

5. Memory and storage allocation. Blocks and programs organized as in the preceding paragraph, shall be stored in core or mass storage on a per user basis, to enable large transfers between the front-end and the host, and thus reduce the number of interrupts between the two units.

Fail Soft

In the case of host failure the users must be advised that the system is no longer available and that they may or may not continue to use it during the down-time. While the front-end provides for an orderly shut-down of the system, transmission may be allowed to continue despite host failure. When the host is returned to service, the front-end should be able to orderly restart itself and place the whole communications network in ready condition.

Application Oriented Functions

The front end should have the capability to be programmed to perform functions such as message switching, data concentration, message logging and traffic recording, and several other functions which may be particular to the system.

V. SOFTWARE REQUIREMENTS

General

Many of the statements made in the previous section will be repeated here, but from the software viewpoint. The software used to support the operation of the system can be regarded, as the adhesive which binds the functions performed by the front-end, in one efficient and coherent system.

The technical parameters to consider in selecting software depend on its ability to (a) manage the sending and receiving of data over the network, and at the same time constantly monitor and discipline this environment, (b) the facility to handle a mix of terminals with varying characteristics and disciplines, (c) to handle varying volumes of data with relatively uniform performance, (d) to perform above present-day speed and volume requirements. With a view to future growth and requirements, a reliable source of software support should be found.

The software requirements of the system can be subdivided into message control and message processing. Message control refers to software which defines the communications environment, establishes the detail of line control, and polices the handling and routing of messages between the computer and the remote terminals.

Each of the major divisions can be broken down further into small software modules. This breakdown provides the ability of using large numbers of permutations and combinations of modules to meet a large variety of application requirements. The major modules usually required by most systems are presented in the following paragraphs.

Line Control

Line control programs will be made up by the following:

1. Call origination unit. This unit will provide for call origination over dial-up, dedicated point-to-point, or multipoint networks. For the case when dial-up facilities are used, EIA standard recommendation RS - 366, provides detail of control signals interchanged between the terminal equipment, in this case generated by the software of the front-end, and the data communications equipment, such as automatic calling units, modems and the voice/data switching equipment supplied by the common carrier. In the case of dedicated circuits, the signals available for establishing connections are in accordance with EIA standard recommendation RS-232. In addition to this the software provides signals as required to activate the proper remote terminals. Multipoint and multidrop circuits

require addressing and function codes to place the desired terminal in the required mode.

2. To answer calls the software must also interpret signals which appear at the front-end interface with the communications equipment. For example activation of the Data Carrier Detect or the Ring Indicator signals between the modem and the front-end interface unit indicates incoming traffic. Recognition and response to these signals should be provided by software.

3. Call termination procedures call for software which manipulates interchange signals from EIA RS-232 or RS-366 interfaces. Once such signals are recognized the program will be directed to disconnect the line in an orderly fashion.

One important aspect in this area, which often leads to misunderstandings between the data terminal equipment supplier or user and the common carriers, is the agreement on timing between action and reaction to the control signals. Care must be taken to agree and install hardware and software which eliminates such errors. One useful method to achieve this is to have the communications equipment controlled totally by the terminal equipment.

Link Control

To ensure message integrity it is necessary to establish a procedure which controls the actual flow of data between the front-end and the remote site or vice-versa. The difference between the line and link control procedures is in that the first places and maintains in operation the data communications equipment, whereas the latter acts on the data terminal equipment. There are areas where the two overlap. The American National Standards Institute has issued its sixth draft of ANSI X3S3.4/475 "Proposed American National Standard for Advanced Data Communications Procedures (ADCCP)". This standard defines the meaning of each control character and proposes sequences which make up link control procedures for a number of applications.

The software of the front-end should recognize such control characters and branch to programs to take action as required by the control character. Message control, such as recognition of start or end of messages and blocking of data into physical blocks is also covered by link control procedures. For example the system must be able to issue or recognize Start of Text (STX), End of Message (EOM) characters and branch to a program whereby the received data will be processed.

Data Transfers

The system software must have the capability to transfer data accumulating in the I/O buffers to or from remote terminals in an orderly fashion and at the proper speed. The prime consideration here is to save processor time in handling data transfers. Even though most communication systems are byte or character oriented the software must provide for transfer of data in blocks, and if possible directly to memory with a minimum interrupt rate. In other words schemes based on block transfers by direct memory access appear to be the most common. From what has been said so far, it is evident that the most important aspects of the data transfer function are (a) speed, to enhance efficiency and through-put, and (b) message control, such as storing and organizing messages while going in and out of the system.

The parameters to be considered to enhance speed are basically dependent on the I/O channel linking the front-end processor (or memory) with the communications interface. I/O channels can achieve transfer speed of one-million characters per second (cps) with commonly available equipment. State of the art equipment can achieve speeds of about 2.25×10^6 cps.

Message control is necessary to organize the transferred data in blocks. This will enable transfers by interrupting the CPU to transfer large blocks of characters, rather than interrupting for the transfer of each character. The

assembled message (or blocks) shall be stored in the I/O interface buffers for transfer in or out of the system. The decisions to be made in this area concern (a) the block size, which depends on the speed of the terminal equipment as compared to that of other such equipment connected to the same buffer, (b) the buffering scheme, which depends on the method of control most suitable to the software. Storage and message transfer efficiency are the prime considerations here. Random storage, double buffering and sequential-and-random storage and queuing techniques are the three basic alternatives of buffering.

To determine the magnitude and capability of the buffering system statistical information concerning the traffic from each type of terminal must be collected. Such information should include speed of terminal, and line occupancy in each direction of transmission when communicating with the computer. This information can be used in conjunction with the buffer control scheme to determine the buffer size appropriate for the system.

Data Processing

This function of the front end shall comprise (a) code translation, (b) transmission error control and (c) blocking formating and in general organizing the data delivery to the host processor.

The software should translate received data from source to object code and conversely the transmitted data from the computer object code to source code as required. Such code conversions are usually based on a direct look-up table. For example suppose that a lower case m is to be translated from seven-level ASCII (1101101) to eight-level EBCDIC (10010100). If ASCII to EBCDIC translation is to be made in look-up Table A, then the contents of location 1101101 of Table A should be 10010100.

Transmission error control can be provided by checking and processing received data for parity or other error control technique. If errors are found the program may branch to subprograms to advise the user, or alarm the system as required. Simpler methods such as direct feed-back (echoeing), reverse channel acknowledgement in response to link control characters, etc. are some other methods which can be programmed into the system.

Received characters will be blocked in the buffers, and brought into the system in blocks suitable to the front-end software. It is evident that such blocks do not necessarily correspond with physical blocks as required by the data terminal equipment, or logical blocks as required by the statements of the program. The software of the front-end will block data to suit the data terminal, and the user's programming needs.

Queuing of data in terms of priority for transfers between the front-end and the host computer should also be controlled by the front-end software. This part of the software will be part of the front-end-host interface.

Miscellaneous Programs

Program which may add to the front-end capability as a communication processor may be included in the software package. Such an additional capability may be message switching and its associated features, or the front-end may be connected to other front-ends for network load balancing or computer to computer communications in general.

Operating System

The operating system of the front-end will supervise all functions performed by the front-end. It will generate sequences of subprograms to meet the needs of the system and of the user efficiently. Since the operating system programs are resident, it is important to design software with memory conservation in mind, as well as programming flexibility and ease of change. As mentioned previously these parameters must be traded off against each other. However, with a view to accommodating the future of the system, flexibility and ease of change should be favoured.

The main functions of the operating system will be to (a) handle and process interrupts, and supervise the sequence and execution of programs for the performance of communication functions, (b) take proper action to protect the system in case of failure, and (c) protect the security of transferred information.

VI. SYSTEM PROPOSAL

General

There are several commercially available minicomputer systems which can be proposed to be used as front-ends to the SGWU computer system.[6] Having evaluated several systems on paper it was decided that only the Interdata system 55 and the DEC-PDP-11 will meet the criteria specified so far. The former uses two minicomputers, one for performing communications control and the other for data processing. This system is slower and less flexible than the PDP-11, which by virtue of the Unibus gains advantage in speed (2.25 million words per second), and in flexibility by accessing and manipulating data anywhere in the system. The following paragraphs outline some of the points of consideration, for PDP-11 to be used to front-end the CDC-6400 system.

System Description

For the sake of completeness, and at the expense of accurate detailed information a brief description of the PDP-11 system and the components used to make up a front-end will be presented here.

The PDP-11 family includes several types of processors, a large number of peripheral devices and options, and extensive

software. The PDP-11 machines are powerful 16-bit computers which can be cycled at speeds as low as 300 nanoseconds. The CPU contains multiple high-speed, general-purpose registers which can be used to control the Unibus, and to perform arithmetic and logic operations. The memory can be expanded from the basic 4,096 words to 126,976 words in increments of 4,096 words. The memory may also contain up to 32,768 words of solid state memory.

The Unibus is a single, common path that connects the CPU, memory and all peripherals. The Unibus has 56 lines which are used as follows: (a) 16 for data transfers, (b) 18 to contain the address of the transferred data, (c) two lines to control the direction of transmission, (d) one each for master sync, slave sync, parity bit low and parity bit high, (e) 13 priority transfer lines and (f) three miscellaneous control lines. The functions performed by signals on each line are adequately described in the PDP-11 "Peripherals and Interfacing Handbook". The descriptions here will be limited to the point required to demonstrate how the PDP-11 will be adapted to the existing system.

The main modules of telecommunications equipment used to inter-connect the data communications equipment to the front-end and their specification sheets are as follows:-

Asynchronous Line Interfaces Type DC11.

The DC11 series of character-buffered interfaces are used between the PDP-11 and a serial asynchronous line. They can be used to connect the PDP-11 to a variety of asynchronous terminals, or to another computer through a common carrier communications facility. The DC11 has the flexibility to handle many different types of terminals. The line speed, character size, stop-code length, and the data set control lines may be set under program control. Input and output line speeds can be varied independently.

Automatic Calling Unit Interface, DN11.

With the DN11 and a Bell 801 Automatic Calling Unit (ACU) or equivalent, any PDP-11 can dial any telephone number in the Direct Distance Dial Network and establish a data link. The DN11 is a digit-buffered interface, and digits to be dialed are presented as four-bit binary numbers. The interface drives the ACU with EIA -366-C voltages and is connected via a standard 25-pin plug.

Synchronous Interface, DP11.

The DP11 provides a double-buffered program interrupt interface between PDP-11 and a serial synchronous line.

The DP11 interface offers flexibility. It handles a wide variety of terminals and line disciplines (i.e., line

control procedures and error control techniques). A programmer can vary sync character, character size, and modem control leads. Automatic sync character stripping and automatic idling are also program selectable. While idling, the DP11 transmits the contents of the sync buffer.

The DP11 design provides individual interrupt vectors and hardware interrupt priority assignments for the transmitter and receiver. Interrupt priority is jumper selectable. This feature, coupled with the automatic transmit idle capability, enables dynamic system adjustment to peak message activity. For example, the programmer can temporarily ignore the transmitter if receive activity is high.

The DP11 is a fully character-buffered synchronous serial line interface capable of two-way simultaneous communications. The DP11 translates between serial data and parallel data. Output characters are transferred in parallel from the computer to a buffer register where they are serially shifted to the communication line. Input characters from the modem are shifted into a register, transferred to a buffer register, and made available to the PDP-11 on an interrupt basis. Both the receiver and the transmitter are double-buffered. This allows a full character time in which to service transmitter and receiver interrupts.

Asynchronous 16-Line Single Speed Multiplexer, DM11.

The DM11 is a full or half-duplex 16-line asynchronous multiplexer. All lines operate at a common code length and Baud rate. The unit will operate at speeds up to 1200 Baud for the full 16 lines. The DM11 uses the PDP-11 NPR(DMA) facility to assemble and to transmit characters directly to or from core memory.

Full Duplex 8-Bit Asynchronous Line Interface, KL11.

The KL11 will connect the PDP-11 with full-duplex 8-bit asynchronous serial lines and is used with Teletypes or other terminals. The standard interface provides a 20 milliamp. current loop output and a contact closure input for interfacing to standard Teletype circuits. The KL11 can be modified to interface to local EIA terminals by using the DE11A.

Line Programmable Asynchronous Serial Line Multiplexer DH11.

This unit has program controllable speeds, character size, and parity generation and checking. Each of the 16 lines has a double-buffered receiver whose contents, when full, are scanned into a 64-word (16 bits/word) buffer. The buffer operates on a first-in-first-out basis with the output to the Unibus made via the Next Received Character Register (NRCR). The DH11 is restricted to one character per microsecond because each character requires one microsecond to shift one position in the buffer.

The transmitter is loaded directly from core under NPR control. When the transmitter is loaded and properly conditioned it will generate a start pulse within one sixteenth of a bit length. The start pulse and subsequent data bits which make up the transmitted character are a full bit-time each.

Description of Operation

The design of the PDP-11 as a whole and especially that of the Unibus allows tremendous flexibility and power with regard to the role of the PDP-11 as a communications line controller or as front-end. The power and flexibility of the system stem from the fact that data transfers on the Unibus are made in an interlocked mode, i.e. any two devices on the Unibus may communicate with or without processor supervision. The interface which connects devices into the Unibus is usually equipped with hardware and software controlled functions, which enable the two devices to assume a master or slave position with respect to each other. When this is done the interlocked control signals which supervise the communication of the two devices require that a signal in a slave device is generated in response to a signal from the master device and conversely the master signal is dropped in response to a slave signal.

There are four types of bus data transfer transactions:

1. Data from slave to master (DATI).

2. Same as (a) but inhibits the restore cycle when data is removed from destructive read-out devices (DATIP).

3. Data transferred from master to slave (DATO) in words.

4. Same as (c) but data is transferred in bytes (DATOB).

For an input data bus transaction (DATI or DATIP) the following will take place in the Unibus, and device interfaces:

1. The master sets the control lines to octal zero for DATI and the appropriate slave address.

2. About 150 nsec after it is determined that the bus is inactive the master device issues a master sync.

3. The selected slave at this point, prepares data for transmission to the master.

4. Data is placed on the data lines of the Unibus and a slave sync is issued on the slave sync line of the Unibus.

5. 75 nsec after receipt of the slave sync the master probes the data lines of the Unibus. When the data is accepted the master sync line of the Unibus is dropped.

6. The master at this point may repeat the process if more data is to be transferred or may drop the control lines and relinquish bus control.

7. If bus control is relinquished the master sync line is negated. This causes the slave to drop the data lines and negate the slave sync line.

This operation (DATI) as well as all bus transactions require a minimum of 450 nanoseconds or the Unibus has a maximum transfer rate of 2.2 million words per second. DATIP, DATO and DATOB bus transactions are similar as far as the basic principle is concerned and each of these transactions also requires 450 nanoseconds.

In summary, the first strong point in the operation of the system is the ability of any two devices (mostly communications line interface and memory) to seize the Unibus and communicate with each other asynchronously. In addition the speed of operation, 450 nanoseconds per word or byte transfer, is far above and beyond of transfers possible in other systems.

The ability of each device to seize the Unibus in an efficient manner is due to the power of the PDP-11 priority

scheme. Transfer of bus control from one device to another is determined by arbitration logic, which is part of the processor. Requests for control of the bus can be made at any time (asynchronously) on the bus request (BR) and on the non-processor request (NPR) lines.

The signal sequence by which a device becomes selected as next bus master is the priority transfer (PTR) bus operation. This operation does not actually transfer bus control; it only selects a device as next bus master. The sequence of events is as follows:

1. The device that needs control of the bus asserts the BR (or NPR) line assigned to it.

2. The processor receives one or more BR signals. These signals enter a priority arbitration system, which compares BR levels with the processor priority levels and against the NPR. If a request has the highest priority entering the arbitration system, and the selection acknowledge (SACK) line is clear, the processor asserts the corresponding BG (or NPG) line, to indicate that BR or NPR requests for control of the Unibus were granted. NPG is asserted during the current bus transaction, while BG is asserted only at the end of the current instruction.

3. Each device on the asserted BG line passes the BG signal, unless it is requesting bus control.

4. The first device on the line which has BR asserted responds to the BG by asserting SACK, blocking the BG signal from following devices, and clearing BR.

5. The processor receives the SACK signal and clears BG, (If SACK is not received within 10 usec, time out occurs and the bus grant is cleared automatically by the processor.)

6. The current bus master completes a data transfer and clears bus-busy (BBSY) at the same time it clears the Address and Control lines.

7. The selected device, which is the new bus master, asserts BBSY when BBSY, BG, and SSYN are clear at the end of the previous data transfer. Interrupt Transaction (INTR) may be asserted at this time, if the new bus master is interrupting.

8. SACK is dropped at the same time INTR is asserted if the device is interrupting. If the device is to transfer data first, the SACK signal is dropped prior to the start of the last bus cycle that the device uses.

9. When the new bus master has completed its last data transfer, it clears BBSY. A new bus master then

takes control of the bus. If no device is selected (SACK is clear), the processor asserts BBSY and continues processing. If, instead of clearing BBSY in a passive release, the device asserts INTR, the processor conducts an INTR bus transaction. This is called active release of the bus.

To signify the end of the transaction and to relinquish bus control the bus-master interrupts the processor.

A device may cause the interrupt operation to occur any time it gains bus control with one of the BR levels. It is usually accomplished immediately on becoming bus master; however, it may follow one or more data transactions on the bus.

1. If immediate interrupt operation is to be initiated, a device which has been selected as bus master asserts INTR and a vector address on the D lines, at the same time that it clears SACK and asserts BBSY. If data transfers occur prior to interrupt then SACK must remain asserted until INTR is asserted. If the device has been making data transfers prior to the interrupt, it should assert through the last cycle.

2. The processor receives the INTR signal, waits 75 nsec for deskew to ensure that all bits of the interrupt

vector address are available, and asserts SSYN when the data is read in.

3. The bus master (interrupting device) receives SSYN and clears INTR, the D lines, and BBSY. This constitutes active release of the bus to the processor.

4. The processor clears SSYN when INTR is cleared, and enters the interrupt sequence to store the contents of the current PC and PS registers and replace them with the contents of the location specified by the vector address.

Programming and Software

The PDP-11 has an instruction repertoire of over 400 hard-wired instructions. This large repertoire in addition to the system's ability to manipulate data in core or peripheral device registers make the PDP-11 one of the easiest systems to program. However, Digital Equipment Corporation have developed and offer to DEC equipment users software for most applications including a communications package called COMTEX-11.

COMTEX-11 comprises modules used to control the transfer and processing of data from the variety of terminals attached to the system, as well as modules which contain routines for servicing interrupt requests from the various types of

equipment connected to the Unibus.

When a terminal requires service, to transmit or receive data from the system, it will request bus-mastership. This creates an interrupt, which can be programmed to access an Interrupt Service Routine (ISR). Once the interrupt has been acknowledged, the Terminal Application Program (TAP) appropriate for the calling device will be called in by System Control Interface Package (SCIP). These Terminal Application Programs (TAP) provide for the actual data transfers, and processing of data (code conversion, formatting, error correction etc.). If ISRs are required in any transaction, SCIP will link the active program with the called ISR.

To initiate line programs the user program is linked with COMTEX-11 via the PDP-11 EMT (Emulate Trap) instructions.

Analysis of Capacity Requirements

The maximum traffic that can be generated by the present system is:

1. Card readers. 1200 cards per minute or 20 cards per second, at 80 characters per card. Card reader traffic is 1600 characters per second (cps).

2. Line printers. 1200 and 1000 lines per minute or 32 lines per second. At 135 characters per line,

line printer traffic is 9000 cps.

3. Time sharing ports. 40 terminals operating at a maximum speed of 30 cps. Low speed traffic is 1200 cps.

4. Synchronous 4800 baud lines. Three lines (with 8 bits/character) generate 600 cps each if operating half-duplex, or 1200 cps each if operating full-duplex. Total synchronous full-duplex line traffic expected: 3600 cps.

5. Miscellaneous traffic from terminals such as graphic displays, 200 user terminals can be expected to be of the order of 1000 cps.

If all terminals connected to the system operated at once at their highest speed, with 100% line occupancy during transmissions and in full-duplex mode when possible, the maximum expected traffic would be 16,400 cps. If the system is to triple in the next few years, the maximum expected traffic load for the system will be about 50,000 cps.

In actual working conditions, it is highly improbable that all equipment is active and transmitting or receiving simultaneously. To allow for slow-typing students, lack of volume etc., one could safely assume that the traffic at any time never exceeds 50% of the theoretical maximum. In other words, it is contended here that it will be safe to

design the system for a total traffic of 25,000 cps.

This traffic allows the front-end about 40 usec/character. This time will be used for (a) transferring the character between the line and the front-end core, (b) checking its parity and translating it to the host's object code (c) format and organize received data into programming entities suitable for use by the host computer (d) transfer each assembled entity into the host computer and (e) performing control function associated with each of these steps.

The PDP-11 is so organized that items (a) to (d) inclusive, require one instruction each. Item (e) comprises large and complicated programs such as line interface and modem control, Unibus control, interrupt handling to initiate or terminate each of these control functions, and recognizing characters for link control and/or terminal control.

The percent usage time of the system is then estimated on a per second basis as follows:

1. For items (a) to (d) inclusive there will be four instruction per character. The average instruction time, on the slower PDP-11's, is about 2.5 usec. Therefore items (a) to (d) require 10 usec per character or about 1/4 seconds, or 25%.

2. Line-interface and modem control programs may be up to 100 instruction or 250 usec. per line routine.

If there are ten call origination or termination routines per minute, this task adds only about 40 usec to the overall operation.

3. Unibus control and associated interrupt handling timing requirements can vary depending on the number of characters handled each time. If DH11 multiplexers are used, such routines will be required once each time it is required to empty the 64 character receive buffer, or to fill the 16 output registers. Assuming that receiving buffer alarms are set for about 80% fill, receiving buffers will be emptied each time 50 characters are received. If of the 25,000 cps, 10,000 are inbound traffic, 200 unibus control and interrupt service routines will be created per second. If the transmitted traffic is 15000 cps, and there are only 5 lines receiving in each multiplexer at any time the transmitted traffic will be in groups of five characters, or there will be 3000 routines per second. Each such routine requires about ten instructions, for a total of 32,000 instructions or 80 milliseconds per second.

4. Link control and or terminal control routines are generated in recognition of special characters. It is expected that an average of five such characters

shall be received for each block of data. If the average block size is about 40 characters, there will be 625 blocks/second or 3125 control characters per second. If there are about ten instructions required to service each character and 2.5 usec per instruction about 80 milliseconds/second will be required to perform this function.

Performance of functions directly associated with data transfers require a total of 410 msec/sec or about 41% of the front-end power. This figure is rounded-off to 50% to allow for handling EMT commands and system household functions. It is emphasized that this figure was calculated on the basis of using DH11 multiplexers for connecting the front-end to the telecommunications lines. Furthermore, the number of lines for which the estimate is made is assumed to be 120 low-speed and 18 high-speed.

Hardware Requirements

Digital Equipment Corporation literature contains block diagrams for several typical equipment configurations used in computer-controlled communications systems. The basic hardware components in each of these suggested systems are the PDP-11 processor, the core memory, the Unibus and the line interface units.

The proposed system will comprise the following sub-

systems:

1. One PDP-11/30 or PDP-11/45 processor.

2. One Unibus system equipped with one DB11-B Bus Buffer for extension beyond 20 unit loads if required.

3. Eight DH11-AA Asynchronous 16-line Programmable multiplexers equipped as required with DM11-DX line adapters.

4. Twelve DU11-DA synchronous line interfaces.

5. Core memory of 32,000 16-bit words. The core memory shall be made up by eight MM11-F(4K) modules. This will provide inter-leaving facilities resulting to possible access and cycle times of 400 and 490 nsec respectively. (For transfer of data out of memory 800 nsec memory cycle time will be required).

6. PDP-11 to CDC-6400 data channel interface to connect the CDC data channel to the Unibus. This will allow the front-end to access and use the host-computer and its peripherals via the CDC-PPU.

Software and Core Requirements

The software required to support the operation of the front-end will be resident in the PDP-11 core. In this section, the core requirements for software and data in transit through the front-end is estimated.

SCIP Module for Controlling 120 Lines

This will require 2500 words of core to accommodate the basic SCIP (for 1-TAP and 1-ISR) and SCIP for an additional 120 ISR and TAP modules for an additional 120 lines.

Terminal Application Programs

120 teletype terminals. This requires 1,350 words of memory for the first module and 3,000 for the remainder.

Synchronous lines. Nine 4800 baud lines and three 2400 baud lines simulating type 200 terminal units, will require 2,000 words for the first TAP, and approximately 100 words for each added line, for a total of 2800 words of core.

The program which will handle communications with the PPU will require 600 words of core memory.

Interrupt Service Routines

The ISR for the first DH11 multiplexer shall require about 600 words of core for its own control program and

approximately 300 words for modem control. The ISR for the control program of each additional DH11 and associated modem control will require about 200 words of core. The total core requirement for DH11 ISRs is 2300 words.

ISRs for DU11 interfaces will require about 1300 words for the first unit and 200 for each additional unit for a total of 2700 words.

The ISR for communication with the host computer is estimated to require approximately 500 words.

User Programs

This will have the largest core requirement especially if the system is loaded with features such as several types of polling, automatic calling and automatic answer, selectable rates, special modem control sequences etc. 10,000 words of core will be allowed for user programs.

Assembler

If the system is programmed using the PAL11-S relocatable assembler and LINK11-S loader, 8,000 words of core memory will be required. It will be assumed here that the front-end will have access to the host assembler and only 1,000 words will be allowed to call it.

Traffic Data

If the lines were operating at their maximum rates, simultaneously and if the front-end were to format data in blocks, at any time the front-end would have in core approximately 180 blocks. Since the system has been assumed to operate at 50% of the theoretical maximum the number of data blocks in core is assumed to be 90. If each block is assumed to contain an average 60 characters, some 2,700 16-bit words of core will be required to accommodate the data in transit through the system.

In summary the core requirements will be as follows:

1.	COMTEX-11		
	SCIP	2,500	
	TAP	7,750	
	ISR	5,500	15,750
2.	User Application Programs		
	UAP	10,000	10,000
3.	Assembler Facility (Loader)		
	PAL11-S and LINK11-S	1,000	1,000
4.	Traffic Data		
		2,700	2,700
TOTAL CORE REQUIREMENTS			29,450

VII. INTERFACE WITH THE HOST.

The PDP-11 will interface with the PPU by connecting the 6000 data channel to the Unibus. This will be done by means of a special interface. The following paragraphs outline some of the desired characteristics of that interface.

Hardware Requirements

Signal Line Alignment

To match the CDC type 6000 data channel to the Unibus, the interface will be required to:

1. Activate the proper lines in the channel and the Unibus to control the direction of transmission. This is necessary because the channel operates full-duplex and the Unibus half-duplex. The Unibus transmission control lines are set to positions 00 and 01 for data in (DATI) and positions 10 and 11 for data out (DATO). These settings correspond to the data channel being in the output or input mode respectively.

2. Set Unibus data lines 12-15 to logic 0. Through-connect Unibus data lines 0-11 inclusive to the corresponding lines of the input and output paths of the data channel.

3. Connect the active flag for both the input and output sections of the data channel to one of the bus request lines. Activation of these lines will initiate a bus request. When the request is granted, the BG line may be connected through the interface to set an interrupt on the data channel and provide the PPU with a status word indicating that the interface is now bus master. Acknowledging this interrupt the interface sends SACK signal to the PDP-11, and sets the BBSY line.

4. Connect the Master sync and the slave sync lines of the Unibus to the full and empty flags respectively for the input section of the channel, and to the empty and full flags respectively for the output section.

5. Connect the master clear (MC) line to the initialization control line of the Unibus. This will clear the Unibus side of the interface when MC occurs.

The preceding Unibus-6000 data channel line alignment was decided on the basis of having the PPU assume mastership of the Unibus in all cases.

Time Base Alignment

The basic time element in the PPU is one minor cycle

(100 nsec). The clock rates available to the channel are 1MHz and 10MHz. The width of the clock pulse, and of the data and control signals is 25nsec. (The MC is a 1 usec pulse repeated every 4096 usec while the DEAD START switch is on). Each data channel can handle a maximum rate of one word per microsecond.

To allow for data transfer skews all signals travelling on the Unibus must have a minimum duration of 75 nsec. Since transfers on the Unibus are asynchronous, the maximum duration of these signals is determined by the rate at which they are sampled.

To ensure proper timing coordination, signals in the PPU to PDP-11 direction must set flip-flops at the interface, which will be cleared by strobbing signals on the Unibus. Conversely signals in the Unibus to PPU direction will operate single-shot circuits whose output will be a pulse of 25 msec, when sampled.

Since the Unibus is capable of transfer rates of 2.25 million words per second and the data channel one million words per second, the actual transfer rate will be determined by programming which is expected to be slower.

Electrical Alignment

The Unibus signal levels are: logic 0 = 3.4 volts, the logic 1 = 0.8 volts and the switching threshold is 2V.

The CDC-6000 data channel signals are 8V pulses of 25 nsec duration. Circuitry to ensure electrical compatibility will be required.

Programming

Sequence Description

To incorporate the front-end to the system without many changes, it is necessary to retain the instructions, function codes and basic data transfer programs for the CDC data channel and the PDP-11 Unibus. To achieve this, it will be necessary to equip the PDP-11 and the PPU with programming modules, which will be used to interpret each other's function codes in a manner similar to that employed by peripheral devices under the present system. The recommended procedure for data transfers between the PPU and the PDP-11 is as follows:

1. The PPU waits for the channel which is connected to the front-end to become inactive (AJM), and then issues a function code (FAN or FNC) to select the interface. This sets the channel active flag, which is transmitted through the interface to activate the BR line assigned to the interface. When BG is asserted in reply to BR, the interface will interrupt the PPU to provide a status reply, and will signal the PDP-11 to

acknowledge receipt of the BG signal. If the status reply is affirmative, the PPU will proceed with its handshaking procedures with the PDP-11.

2. When the link between the PPU and the PDP-11 is established, a function code may be transmitted to the PDP-11 to activate the proper control line for either input or output, i.e. condition the Unibus for DATI or DATOB transactions. When this is done the front-end via the interface shall provide the PPU with the appropriate status word.

3. If the status word is affirmative, the PDP-11 is placed in the desired mode and it is ready for data transfers. When the PDP-11 is transmitting (DATOB) to the PPU, it will issue a SSYN with each byte to activate the channel full flag through the interface. The PPU will remove the character and deactivate the channel full flag by activating the channel empty flag. The channel empty flag will be converted to MSYN at the interface and will start the next DATOB cycle. When the PDP-11 is receiving, its SSYN pulses set the channel empty flag of the output section of the data channel, and the channel flag of the channel appears as a MSYN on the Unibus. The corresponding instructions executed by the PPU are IAN or IAM for DATOB and OAN or OAM for DATI.

4. When the data has been transferred, more function codes will be transmitted by the PPU to release the PDP-11, to cause the interface to relinquish the Unibus, and release the data channel. In the case of DATOB, the PDP-11 must provide as the last character of the transmitted block a special character to cause the PPU to start the disconnect sequence. Alternately the A register is loaded with the number of words to be transferred before execution of IAM. When this number becomes zero, the path control is relinquished.

Timing.

The programming sequence to establish, maintain and relinquish the link between the PPU and the front-end, requires about 40 PPU instructions or approximately 80 usec per block. If the average block size is about 50 characters, the IAM or OAM instruction will be executed once per character. The IAM and DATOB or OAM and DATI instructions are sequential, and their execution consumes a minimum of 1.45 usecs. Since the operations on the channel are synchronous, (once every major cycle during one minor cycle) IAM instructions must be placed 2.0 usec appart. For a block of 50 characters, and additional 100 usec will be required per block, for a total of 180 usec per block or 3.6 usec per character. This results to an average transfer rate of 278 Kcps.

Software

In order for the front-end to interface with the PPU and the host, several modifications must be made to the PPU programs. The reason for this modification is that the front-end will take over several functions of the PPU programs. Therefore the PPU should retain subprograms required to interface with TELEX and IMPORT-EXPORT in the host, or with the front-end for control of the link and interface between the data channel and Unibus. For example the multiplexer driver (1TD overlayed with 2TD) will have to be relocated from the PPU to the PDP-11. On the other hand overlay 1TA, (TELEX AUXILIARY ROUTINE) which processes TELEX functions requiring PPU action, will have to be retained in the PPU.

Realizability

The preceding paragraphs outline some of the major points of consideration, if the data channel-Unibus interface were to be designed. However, it is believed here, that existing, proven systems should be preferred, if there is a possibility of adapting them with ease. To this end, it is recommended to seek the consultation of people with experience in the design of similar systems. The University of Montreal Department du Calcul have developed an interface for front-ending their CDC system, operating under Scope, to a PDP-11 computer. For a fee they will design the interface, and reorganize the system to accept this front-end.

VIII. COSTS

The cost to purchase and install a PDP-11 computer to front end the SGWU CDC-6400 system in its present state is estimated as follows:

Hardware for Present	
PDP-11 computer equipped with 24K of core	$ 22,000
DH-11 multiplexer for asynchronous lines (3 units)	28,500
DU-11 Synchronous line interface (4 units)	4,000
Miscellaneous, cabling, connectors, etc.	3,000
UNIBUS - Data channel interface, installed, tested and equipped with software for operation, maintenance and testing	11,000
Hardware Installations (Labour)	6,500
Total Hardware Cost	$ 75,000
Software for Present	
PDP-11 Communications Package, COMTEX-11 or Equivalent and Modifications to Existing Software	$ 45,000
GRAND TOTAL	$120,000

Costs to Expand to Full Capacity

Purchase and installation of equipment to expand the front end system to the level provided for in this paper will incur the following additional costs:

Memory	$ 7,000
DH11 Multiplexers (5 units)	$ 49,500
DU11 Synchronous line controllers (10 units)	$ 10,000
Miscellaneous Cables, Connectors, etc.	$ 5,000
Installation Labour and Contingencies	$ 8,000
Total for Expansion	$ 79,500
GRAND TOTAL FOR MAXIMUM SIZE	$199,500

To appreciate the fully expanded system layout with respect to the aforementioned costs, reference should be made to figure 3. For comparison purposes the present system layout is shown in figure 2.

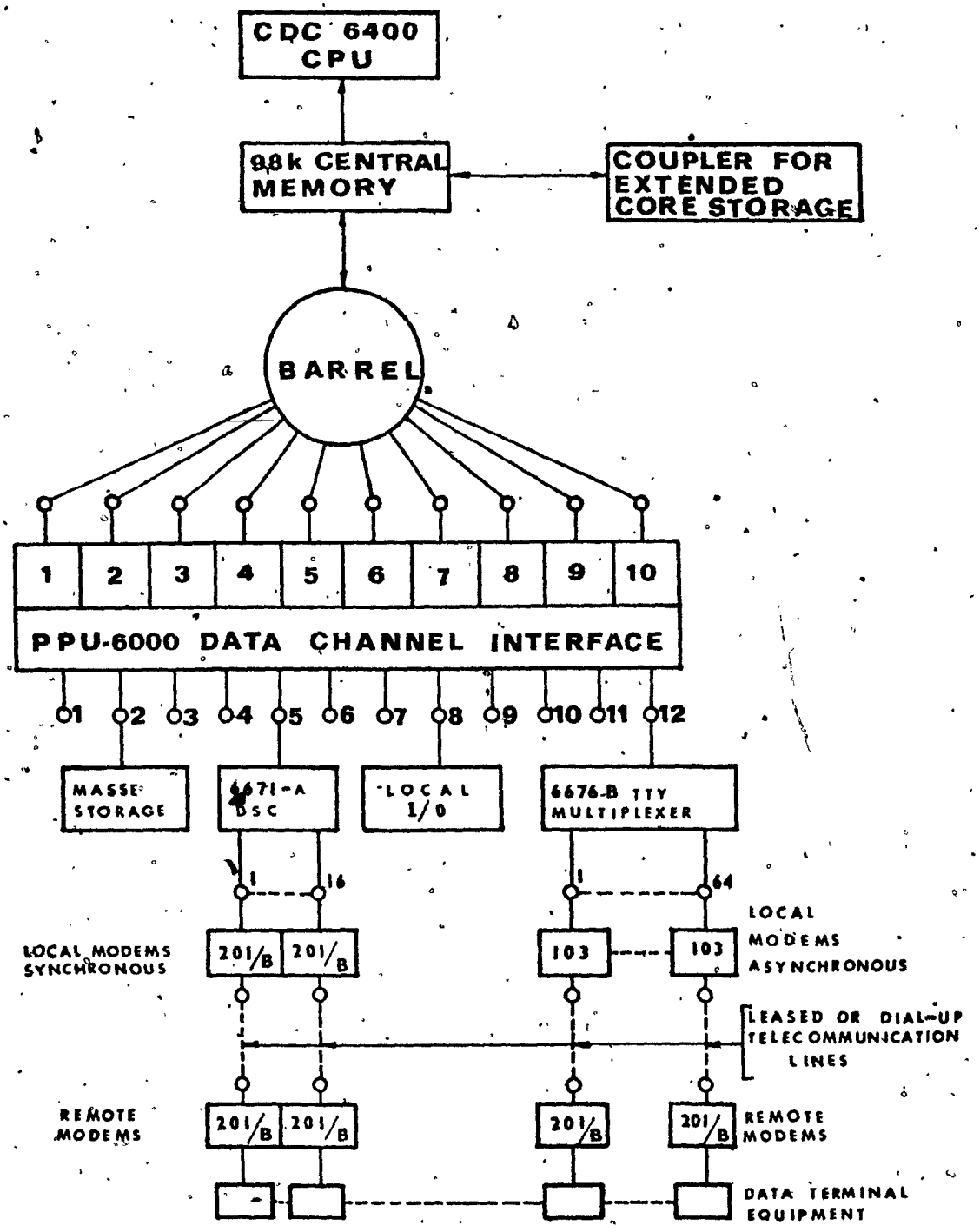

FIG.2 CDC - 6400 COMPUTER SYSTEM LAYOUT

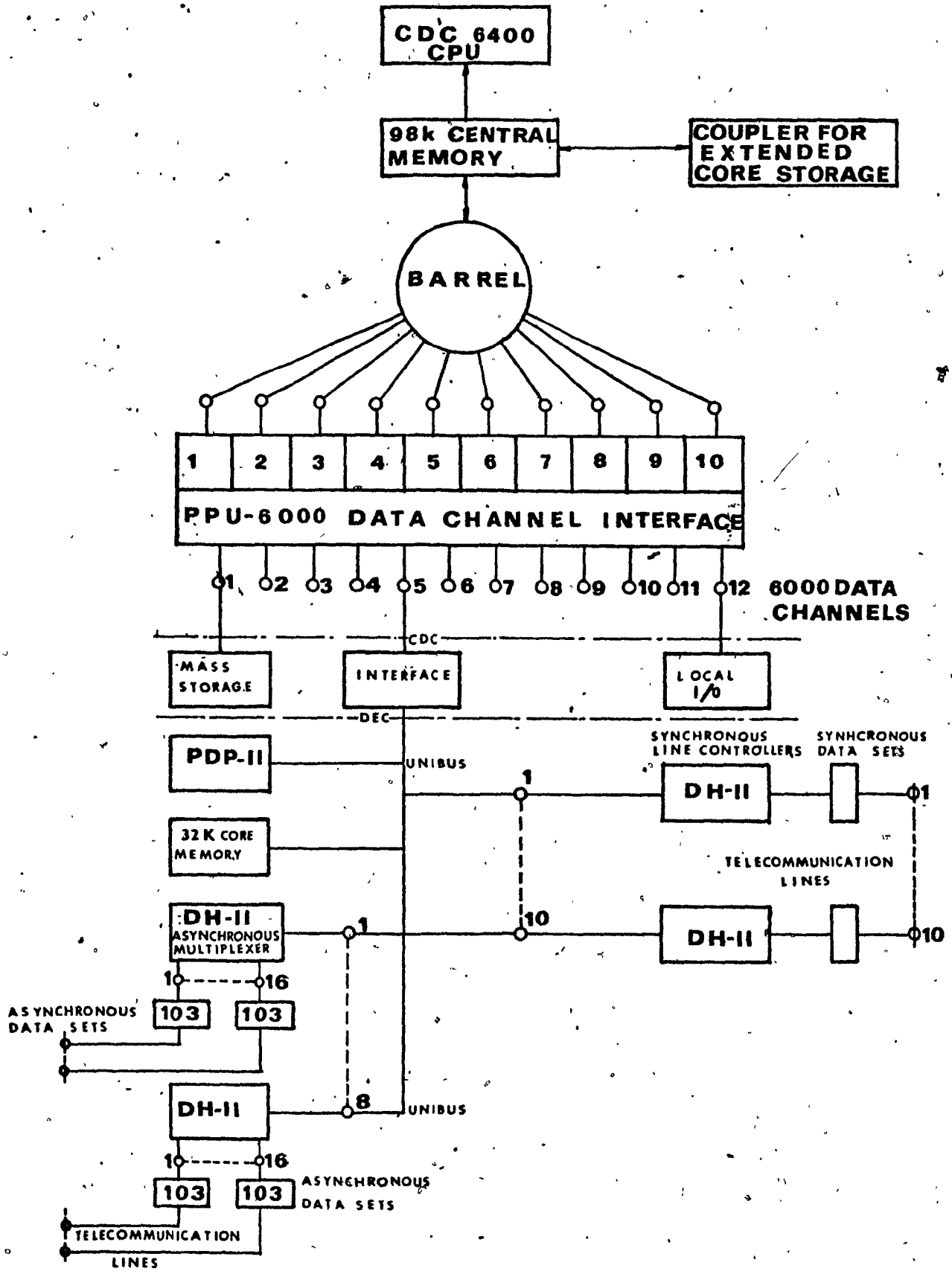

FIG.3 PROPOSED SYSTEM LAYOUT

IX. CONCLUSION

The benefits to be derived from the installation of a front-end system to the CDC computer system are flexibility, power and the basis for expansion more economically than with the present arrangement. These reasons, even though they appear intangible, should more than justify the $120,000 estimated cost to upgrade the present system, and the additional $80,000 required to develop the system to the maximum expected capacity.

LIST OF REFERENCES

1. Stahl, Jack. "Modems - Another Look - Telecommunications Presents the 5th Annual Modem Survey", Telecommunications International, Volume 7, Number 6, June 1973, page 27

2. Parker, Donald W. "Telecommunications Survey - Multiplexer Selection for Data Communications", Telecommunications International, Volume 7, Number 1, January 1973, page 21

3. Telecommunications Staff, "Telecommunications Survey - On-Line Applications for Alphanumeric Printers", Telecommunications International, Volume 6, Number 12, December 1972, page 21

4. Telecommunications Staff, "Telecommunications Survey - The Communications Mini-computer", Telecommunications International, Volume 6, Number 10, October 1972, page 15

5. Telecommunications Staff Report, "Communications Processors - Who, What & Why?", Telecommunications International, Volume 7, Number 5, May 1973, page 19

6. Glasgal, Ralph. "Advanced Concepts in Time Division Multiplexing", Telecommunications International, Volume 6, Number 10, October 1972, page 27

7. Gaines, Eugene C. "Specialized Common Carriers - Competition and Alternatives", Telecommunications International, Volume 7, Number 9, September 1973, page 15

8. Dobbie, James. "The Changing Role of the Computer in Data Communications", Telecommunications International, Volume 7, Number 1, January 1973, page 29

9. Stone, Irwin M. "The Changing Needs of Minicomputer Users", Telecommunications International, Volume 7, Number 1, January 1973, page 37

10. Bazzy, William. "Telecommunications Trade Through the Decade", Telecommunications International, Volume 7, Number 5, May 1973, page 8

11. Walker, Edward A. "Some Guides to Specifying your Communications System", Telecommunications International, Volume 7, Number 5, May 1973, page 29

12. McKee, David J. "How Transaction Cost Declines as Data Networks Get Larger", Electronics - Data Communication Systems, September 1973, page 17

13. Price, Roland J. "Estimating Buffer Size and Queuing Delay in a Remote Concentrator", Electronics - Data Communication Systems, September 1973, page 39

14. Chysic, John F. "Fundamentals of Software for Communications Processors", Electronics - Data Communications Systems, September 1973, page 47

15. Kersey, J. Ray. "Taking a Fresh Look at Data Link Controls", Electronics - Data Communications Systems, September 1973, page 65

16. Wang, Keui-Seng. and Dalyai, Stephen A. "Don't Overlook Electrical Compatibility When Mating Equipment", Electronics - Data Communications Systems, September 1973, page 73

17. Deal, Richard L.; Chysik, John.; Markus, Julius.; and Jennings, Richard. "Datacomm Minis: Special Versus General Purpose", The Data Communications User, October 1973, page 21

18. The Data Communication User, "Communications Software and Networks", The Data Communication User, December 1972, page 14

19. The Data Communications User, "Communications Processors", The Data Communications User, October 1972, page 16

20. The Editors of the Data Communications User, "The User's Guide to Selecting Data Communications Equipment, Systems and Services", The Datacomm Planner, August 1973, All Articles

21. Secretariat Business Equipment Manufacturers Association, "American National Standard Procedures for the Use of the Communication Control Characters of American National Standard Code for Information Interchange in Specified Data Communications Link", American National Standards Institute, Inc., March 10, 1971, ANSI X3.28-1971

22. Auerbach Data Communications Reports, "User's Guide to Communications Processing Equipment Reports", Auerbach Data Communications, User Guide 8000:01, October 1965

23. Auerbach Data Communications Reports, "Search Chart: Communications Processors", Auerbach Data Communications, Report No. 7715.0000.010, September 1972

24. Auerbach Data Communications Reports, "Comten 65/45/20 Communications Processors", Auerbach Data Communications, Report No. 7715.2898.040, December 1972

25. Auerbach Data Communications Reports, "Interdata New Series Communications Processors", Auerbach Data Communications, Report No. 7715.4225.020, November 1972

26. Auerbach Data Communications Reports, "IBM Corporation 2701 Data Adapter Unit", Auerbach Data Communications, Report No. 7715.4239.010, March 1972

27. Auerbach Data Communications Reports, "IBM Corporation 3705 Communication Controller", Auerbach Data Communications, Report No. 7715.4239.040, March 1972

28. Control Data Corporation, Control Data 6000 Series Computer Systems Reference Manual, following sections: System Description, Peripheral and Control Processors, 6681-A/B Data Channel Converter, 6684-A Data Channel Converter, 6676-B TTY Multiplexer, 6671-A Data Set Controller, 6673-A/6674A Data Set Controller, Control Data Corporation, 1973

29. Interdata Inc., "Model 50/55 Communications Processor Reference Manual", Interdata Inc., Publication Number 29-249R01, May 1972

30. General Electric Communications, "Technical Information: Diginet 1600 Data Communications Processor", General Electric - Telecommunications Products Department Document No. EDC-95

31. IBM Corporations, "IBM 2701 Data Adapter Unit: Original Equipment Manufacturer's Information", IBM Systems Reference Library, File No. S 360-19, GA 22-6844-3, June 1969

32. Digital Equipment Corporation, "PDP11 Processor Handbook", Digital Equipment Corporation, 1971

33. Digital Equipment Corporation, "PDP11 Peripherals and Interfacing Handbook", Digital Equipment Corporation, 1971

34. Digital Equipment Corporation, "PDP11 COMTEX-11 User's Guide", Digital Equipment Corporation, 1971

35. Martin, James. "Teleprocessing Network Organization", Prentice Hall, 1970