FX 4-1052 BI
July 2, 1964

Technical Proposal

ELECTRONIC COMMUNICATION CONTROLLER

Prepared for

Sandia Corporation
Albuquerque, New Mexico

LIBRASCOPE GROUP
GENERAL PRECISION
SURFACE EQUIPMENT DIVISION
808 WESTERN AVE • GLENDALE 1, CALIFORNIA

CONTENTS

ILLUSTRATIONS

Section 1

INTRODUCTION

This document presents a proposal to the Sandia Corporation for a Librascope L-1192 Controller-Processor system to provide the functions of a computerized communication controller (RFQ-Reference JMH/4114). The equipment was designed specifically for communication and message switching applications and permits a flexible mix of communication channel speeds, codes, and formats. A stored program approach, coupled with some special instructions and hardware, offers both efficiency and versatility in meeting existing or changing code translation, validity checking, message processing, and editing requirements. The modular design of the Librascope system permits the installation of only those items of equipment necessary to meet present specifications of the Sandia Corporation's telecommunication network. This configuration is field expandable to fulfill future requirements. The L-1192 Controller-Processor has been designed to the same military specifications for quality control and construction as the 473-L Command and Control System, now in production at Librascope. The proposed equipment contains the qualities that have satisfied 473-L requirements for high reliability and non-interrupted service.

The proposed Librascope system is designed to interface with existing or on-order IBM computers and peripheral equipment at Sandia, and some additional IBM devices are required. The proposed additional Librascope and IBM equipment, and a table of major units of IBM equipment assumed to be already available or on-order at Sandia, are listed below. Librascope will assume complete responsibility for obtaining and installing the additional required IBM equipments and interfacing them with the Librascope equipment.

1.1 EQUIPMENT AVAILABLE OR ON-ORDER AT SANDIA

Quantity	Item	Manufacturer
1	1460 Computer	IBM
1	7080 Serial Input/Output Adapter (interface 1460/1192 communication)	IBM
1	7090 Computer	IBM
1	1302 Disk File (shared by 7090 and 1192)	IBM
1	7631 Model 2 File Control (Remove from 7090 and locate with 1192)	IBM

1.2 PROPOSED ADDITIONAL EQUIPMENT

Quantity	Item	Manufacturer
1	Line Unit Console (Containing required interface modules as determined by number of simplex, half-duplex, and full-duplex lines)	Librascope
1	L-1192 Controller-Processor	Librascope
2	Core Memory Console (8,192-words/32,768 char. ea.)	Librascope
1	1301 Disk File (on-line to L-1192)	IBM
1	7631 Model 4 File Control (Permit 7090/1192 shared access to 1302)	IBM

Section 2

LIBRASCOPE EXPERIENCE

Many data processing systems in the past have had the requirement to interface with communication lines. For the most part these involved only slow speed asynchronous teletype devices and employed manual error control and channel coordination procedures. Librascope delivered a controller to process such lines for an early application. This device had a capacity of 12 duplex lines operating at 75 baud, contained wired logic, and utilized a magnetic drum as a data buffer. The primary disadvantage of the unit was that changes to line codes and message formats could only be accommodated by costly and time-consuming hardware changes.

Librascope then developed a stored-program Controller-Processor to meet the need for a more flexible communication buffer. This equipment has been incorporated into the AN/FYQ-11 Data Processor Set of the U.S. Air Force Headquarters 473-L Command & Control System, and is now in production at Librascope. The system will shortly be installed in the Pentagon, where its application requires the simultaneous transfer of messages over multiple duplex communication lines as a subscriber to AUTODIN (U.S. Defense Communication Agency's Automatic Digital Network). The bit-serial message transfers must be framed by the Controller-Processor into 8-bit characters, containing both message and control information. Additional acknowledge or reply codes are also interspersed with message transmission. Each 8-bit character contains a parity check, and messages are divided into 80 character blocks - each providing horizontal parity information as well as framing characters. The receipt and transmission of each block must conform to specified channel coordination procedures. These are carried out by the Controller-Processor program, which also performs the format and parity checks and provides the necessary code translation. The problem demands on-line access to mass files, and a data processing capability to perform specified editing, control, and computation functions - including those necessary to insure message protection and accountability.

The Librascope effort on 473-L and additional extensive application studies on AUTODIN and other message store and forward applications has resulted in the development of a flexible and powerful communication controller. The L-1192 Controller-Processor's stored program provides character and variable-field operations, and will accept a wide range of codes and formats for both asynchronous and synchronous transmission. The communication interface is modularly expandable, to a maximum of 128 simplex, half-duplex, or full duplex channels. The L-1192 Controller-Processor is capable of completely independent operation, or may also provide on-line communication with additional and perhaps dissimilar computer equipment.

Section 3

SANDIA REQUIREMENTS

Librascope personnel have studied the Request for Quotation from the Sandia Corporation for an Electronic Communication Controller. A recent visit to Sandia at New Mexico clarified the status of existing or on-order computer equipment, and provided additional information on application specifications. The following listing presents a brief summary of the Librascope understanding of the Sandia Corporation's requirements for an improved teleprocessing system.

1. Communication Lines

   The system must handle a minimum of 40 multi-drop lines, field expandable to 80 lines. These include simplex, half-duplex and full-duplex channels. (See Sections 4.0-4; 5.1)

2. Line Speed

   Bit rates on the communication channels may vary between 10 and 2400 bits per second. (See Sections 4.0-2; Appendix A)

3. I/O Flexibility

   The central controller must be able to handle a wide mix of message codes and formats, and permit flexibility in desired code translation. (See Sections 4.0-3, -8; 6.4.1.2; Appendix A)

4. Processing Capacity

   Peak activity will be a total of 10,000 input/output characters per minute. A minimum internal memory capacity of 60,000 characters is required. (See Sections 4.0-7, -10; 5.2)

5. Instruction Efficiency

   The instruction set must be capable of efficiently providing data editing and validity checking operations, in addition to meeting standard communication requirements. (See Sections 4.0-11; 5.3.1.2; Appendix B)

6. Random Access File

The controller must provide a minimum random access file of about 20 million characters, including at least 10 million characters of on-line storage. (See Sections 4.0-13; 5.4.1)

7. Controller Interface

The central communication controller must be capable of stand alone operation, and must also permit:

a. Two-way communication with an IBM 1460 computer. (See Sections 4.0-14; 5.4.3)

b. Two-way communication with an IBM 1302 Disc File, which is shared by an IBM 7090 Computer. (See Sections 4.0-15; 5.4.2)

8. Operator/Maintenance Console

The central controller must provide a control panel and/or console to facilitate diagnostics and corrective action. (See Sections 4.0-11; 5.3.4)

9. Dial Facilities

The central controller must be adaptable to dial up out and dial answering facilities. (See Sections 4.0-5; Appendix A.5)

10. Modularity

Modularity of equipment design is an important characteristic and requirement for this system. (See Sections 4.0-16; 5.0)

11. Reliability

Data protection, system recovery, and message accuracy are of primary importance. Equipment reliability in terms of high MTBF and low MTR is essential. (See Sections 4.0-17, -18, -19; 5.3.1.5; 6.0)

12. Software

Information on support software and methods for loading programs is required. (See Sections 5.3.4.5; 7.0; Appendix C)

Section 4

SUMMARY OF APPROACH

The Librascope L-1192 Controller-Processor system is proposed to process the telecommunication requirements of the Sandia Corporation. This equipment provides both single character and multi-character field processing capability. It offers an input-output flexibility for handling a wide mix of communication speeds, codes, and formats for both synchronous and asynchronous transmission. The basic organization of the proposed system is presented in Figure 4-1, and the principal features are listed below. More detailed equipment descriptions and available programming techniques are provided in the following sections and in the Appendix.

1. Line Interface

   Each input or output channel for a communication line terminates in either an asynchronous line interface module or in a synchronous line interface module. Up to 64 input and 64 output interface modules are contained in a Line Unit Console.

2. Line Speed

   An asynchronous interface module will handle TTY and other required codes (e.g. ASCII, BCD, Friden, Four of Eight, etc....) at up to 150 bits/second. The synchronous interface module will accept the required data codes at up to 4800 bits/second.

3. Line Flexibility

   Asynchronous and synchronous interface modules may be added, replaced, or interchanged in the field, as required.

4. Communication Interface

   The interface modules communicate directly with the L-1192 Controller-Processor, and buffer the input or output data. The central controller may connect up to two Line Unit Consoles, to service a maximum of 128 input and 128 output interface modules. A simplex line requires either an input or an output interface

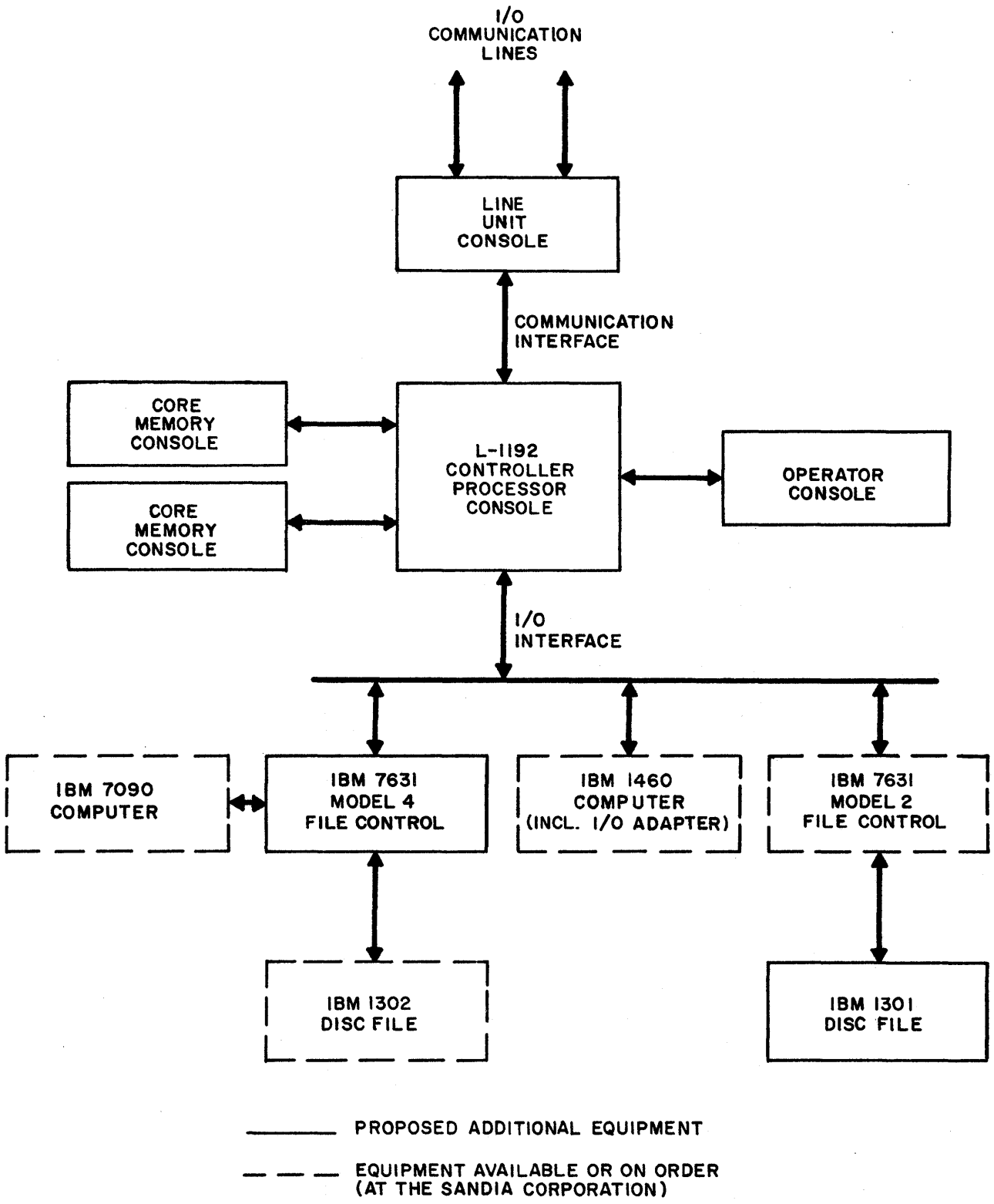


Figure 4-1. System Configuration

module, while a half-duplex or full-duplex line requires an input-output pair of modules. The number of actual interface modules provided with a Line Unit Console is dependent on the number and type of operational lines and the desired number of spare units in the communication controller system. The Communication Interface will also control up to 16 on-line devices, such as typewriter and paper tape read/punch equipment.

5. Dial Facilities

The L-1192 Controller-Processor has been designed to make it adaptable to dial up out and dial answering facilities. The circuit switch is controlled by data transferred through the standard line interface modules. A recent Librascope design interfaced a Controller-Processor with an AUTOVON dial switching system (Appendix A. 5), but specific equipment requirements and program procedures are dependent on the particular specifications of the designated dialing facility.

6. Central Controller

The L-1192 Controller-Processor is a stored program computer, and utilizes standard 8, 192 word core memory modules with a 5. 0 microsecond cycle time. Faster core memory speeds are also available.

7. Memory Capacity

A central controller word provides 32 data bits plus one parity bit, and contains either a computer instruction or four 8-bit characters. The 32-bit word format is well suited to character processing, indexing, and addressing. Two core memory modules, providing a total of 16, 384 words (65, 536 characters) of storage with a 5. 0 microsecond cycle time are proposed for the Sandia application.

8. Code/Format Flexibility

The L-1192 Controller-Processor has been designed to efficiently and economically process communication requirements. Care has been exercised, however, to permit flexibility of operation

through stored program. Changes or addition of acceptable character codes may be accommodated by the simple revision of stored translation tables. New procedures or requirements for the editing, formatting, or validity checking of message data may also be implemented by modification to the stored program.

9. Instruction Efficiency

The input/output communication data automatically transfers between assigned areas in core memory and the line interface modules. Input data and output requests are efficiently accessed by a special-purpose instruction. The remaining L-1192 instructions are more general-purpose in nature, and provide both single character and multiple-character binary, decimal, and alphanumeric field operations. A group of eight data/arithmetic registers remove the bottleneck normally encountered with a single accumulator. Almost all the L-1192 instructions permit both indirect addressing and indexing.

10. Processing Capacity

The proposed L-1192 Controller-Processor system will provide the standard communication functions for a maximum input plus output rate of about 3000 characters/second. These programmed operations meet AUTODIN specifications for message validity and parity checking, channel coordination, code translation, block formatting, and message storage and retrieval. Any additional editing or special processing requirements of the Sandia application must also be considered in determining the actual processing capacity of the L-1192 System (see Section 6.0, Programming Techniques).

11. Operator/Maintenance Console

A control panel, a 300 character/second paper tape reader, a 60 character/second paper tape punch, and a 15 character/second input-output typewriter are provided to assist in program assembly and debugging, and to facilitate equipment maintenance/diagnostic action.

12. I/O Interface

The L-1192 Controller-Processor may communicate with up to 16 peripheral devices via an I/O Interface Channel. The proposed system will communicate with an IBM 1301 Disk File, an IBM 1460 Computer, and share IBM 1302 Disk File access with an IBM 7090 Computer. Service Request and End of Operation program interrupts are provided. A single interface transfer can proceed at any one time, and will overlap both central controller and communication interface operations.

13. IBM 1301 Disk File Interface

The L-1192 Controller-Processor will store and retrieve messages on an IBM 1301 Disk File at a transfer rate of approximately 70,000 8-bit characters/second, or 90,000 6-bit characters/second. The IBM 7631 Model 2 File Control Unit now utilized by Sandia with the IBM 7090 Computer will interface between the L-1192 and the 21,600,000 8-bit characters or 27,960,000 6-bit characters of on-line random access storage provided by the 1301.

14. IBM 1460 Interface

The L-1192 Controller-Processor will provide two-direction communication directly with the IBM 1460 Computer System on-order for Sandia. The transfer may proceed at speeds up to the maximum IBM 1460 data rate, via the IBM 7080 Serial Input/Output Adapter.

15. IBM 7090 Interface

The L-1192 Controller-Processor will communicate with the present IBM 7090 Computer System through shared access to the IBM 1302 Disk File. An IBM 7631 Model 4 File Control Unit will be necessary to interface and switch the IBM 1302 between the L-1191/IBM 7090. The transfer rate will proceed at 143,000 8-bit characters/second or 184,000 6-bit characters/second.

16. Modularity/Expandability

The L-1192 Controller-Processor can control up to 128 duplex communication lines. Line interface cards are added only as needed, and each group of 16 line interface terminations are contained in an independent Line Group unit. Core memory is expandable in 8, 192-word modules, with a maximum capacity of eight modules or 65, 536 words. A maximum of 16 I/O Interface Channels may be connected to the I/O Interface of the L-1192, and up to 16 peripheral devices serviced by each channel. A total of 5 IBM 1301 or 1302 Disk File Units may be serviced by each IBM 7631 File Control Unit. A maximum of 16 typewriter, paper tape reader/punch and similar on-line devices may be connected to the Communication Interface.

17. Equipment Reliability

The L-1192 Controller-Processor has been designed in accordance with military specifications for quality control (MIL-Q-9858) and construction techniques (MIL-P-11268), assuring unusually high component and equipment reliability. The same qualities that have satisfied the previously mentioned 473-L requirement for non-interrupted service are contained in the proposed configuration.

18. Data Protection/Accuracy

The L-1192 Controller-Processor provides internal error checks to insure data protection. These include hardware tests for illegal operation codes, in addition to parity checks on transfers to and from core memory, the IBM 1460, and the IBM 1301/1302 Disk File Units. Parity checks of input characters, as well as other message format and validity checks, are executed by the program. The operational program for the central controller must also maintain a ledger balance between incoming and outgoing message records, to provide a means of determining that every message accepted by the teleprocessing system has been appropriately handled.

19. System Recovery

System restart and recovery routines must be incorporated into the design of the operational program to permit the system to regain operational capability following communication line or central controller equipment malfunction. The normal procedure to meet this requirement has been to retain a copy of all message increments and control information on an independent storage media. Further, controller assigned identifying numbers are used to permit on-line audit of correctness of sequence and completeness in handling of message segments and complete messages. The recovery routines perform functions such as retrieval of required messages from reference storage, re-establishment of recovered messages in queues, and protection of message accountability.

20. Support Software

The software package offered with the L-1192 system includes an assembler, a disc I/O control routine, a loader system, a memory dump routine, and an equipment acceptance test program. These are discussed in Section 7 of this proposal and in the Appendix.

21. Operational Support

Librascope personnel have extensive experience in processing application requirements similar to the proposed Sandia communication teleprocessing system. Some operational programming techniques are presented in Section 6 of this proposal. Qualified personnel will be available to assist the Sandia Corporation in implementing operational requirements, either simply as advisors or under contract to contribute to the systems design and programming effort.

Section 5

EQUIPMENT DESCRIPTION AND OPERATION

This section discusses the purpose, operation, and design of each item of equipment proposed for the Sandia communication teleprocessing system.

5.1 LINE UNIT CONSOLE

The Line Unit Console provides the circuits for interfacing as many as 64 duplex communication lines with an L-1192 Controller-Processor. Each group of eight input and eight output line interfaces are packaged into a Line Group Module, and up to eight such Line Groups may be contained in a Line Unit Console. A Line Group module provides its own power supplies and line interface module scanner, and is plug-in rack mounted. This method of packaging increases system performance by insuring that a power supply failure will affect a limited number of communication lines.

A Line Group may contain up to eight input and eight output interface modules, each providing up to one character of input or output buffering for a communication line. There are four types of interface modules available for use in a Line Unit Console: to receive data on a synchronous line; to transmit data on a synchronous line; to receive data on an asynchronous line; and to transmit data on an asynchronous line. (A detailed description of these interface modules is presented in Appendix A). A simplex communication line will require either an input (receive) or an output (transmit) interface module, while a half-duplex or full-duplex line utilizes an input-output pair within a Line Group. The synchronous line interface modules provide the character framing for the high speed lines, and will operate at up to 4800 bits per second. The asynchronous interface for slower speed lines accumulates standard TTY or other required codes, and will operate on up to 150 bits per second. The line interface modules transmit the input data or the output request to the Controller-Processor, along with the line interface identification number. A slow speed asynchronous channel position in the Line Unit can be converted to

a high speed synchronous line terminal by simply replacing the interface plug-in module.

The allocation of input and output communication lines to specific line interface modules is made by the system Supervisor, who must provide this information to the Controller-Processor. A failure in a line interface module will result in the order to replace the module or to switch the communication line to a spare interface. The number and type of interface modules actually provided with the Line Unit Console is dependent on the number of simplex, half-duplex, and full-duplex operational lines and the number of spare units required by the communication teleprocessing system. Up to two Line Unit Consoles, interfacing a maximum of 128 input and 128 output lines, may be connected to the L-1192 Controller-Processor Console.

5.2 CORE MEMORY CONSOLE

Core memory for the L-1192 Controller-Processor system is provided in modules of 8,192 words, each contained in an independent console. The L-1192 may address up to a maximum of eight modules, for a total of 65,536 words of memory. A word contains 32 data bits plus an additional bit to maintain odd parity. Each data word provides four 8-bit characters, which may be individually addressed by the L-1192 instructions. Transfers to and from core memory are in word-parallel, and these are parity checked. The standard core memory module offers a read-write cycle time of 5.0 microseconds. Two such Core Memory Consoles, providing a total storage of 16,384 words or 65,536 characters, are propsed for the Sandia application. Higher speed 2.0 microsecond core memory modules are available as options.

The core memory provides storage for the operational program, the message buffering, and the required activity, system status, and output tables. A channel activity table (CAT 2) has been assigned 256 word positions in core memory (octal 20 through 417) to accept all input characters and the output requests from slow speed lines. A similar table (CAT 1) is assigned 128 memory word addresses 420 through 517, and

accepts output requests from the high speed lines. Output characters are accessed from a Line Character Buffer Table (LCBT), which has one character position assigned in memory per output line – packed four per word in addresses 420 through 457. The line interface modules operate with the serial transmission on the communication lines, and normally contain either a character assembly (input) or disassembly (output) shift register. These provide the bit-to-character or character-to-bit serial transformations. Input characters and requests for output characters automatically enter the Channel Activity Tables, along with a line identification number. These inputs are available to L-1192 operational program. Output characters are also automatically accessed from the LCBT, and routed to the proper output interface module (see Section 5.3.2). The transfers between core memory and the input/output line interface modules, the on-line devices (typewriter, paper tape), the selected peripheral device (1301, 1302, 1460), and the Central Controller arithmetic and logical unit occur on a cycle stealing basis – permitting overlap of all these operations.

5.3 L-1192 CONTROLLER-PROCESSOR CONSOLE

The L-1192 Controller-Processor Console consists of an Arithmetic and Logical Unit, a Communication Interface, an I/O Interface, an Operator Console with its associated electric typewriter and paper tape equipment, and the Adapters to operate with IBM Disk File and 1460 equipment. The L-1192 Console is presented in Figure 5-1.

5.3.1 Arithmetic and Logical Unit

The L-1192 ALU includes the instruction register and instruction decode logic, the field operand address register, the program counter, and eight individually addressable data registers. It is specifically designed to facilitate the processing of data from a number of communication lines. The machine organization enables operations to be performed on individual characters or on multi-character binary, decimal, and alpha-numeric fields. The block diagram of the ALU is shown in Figure 5-2.

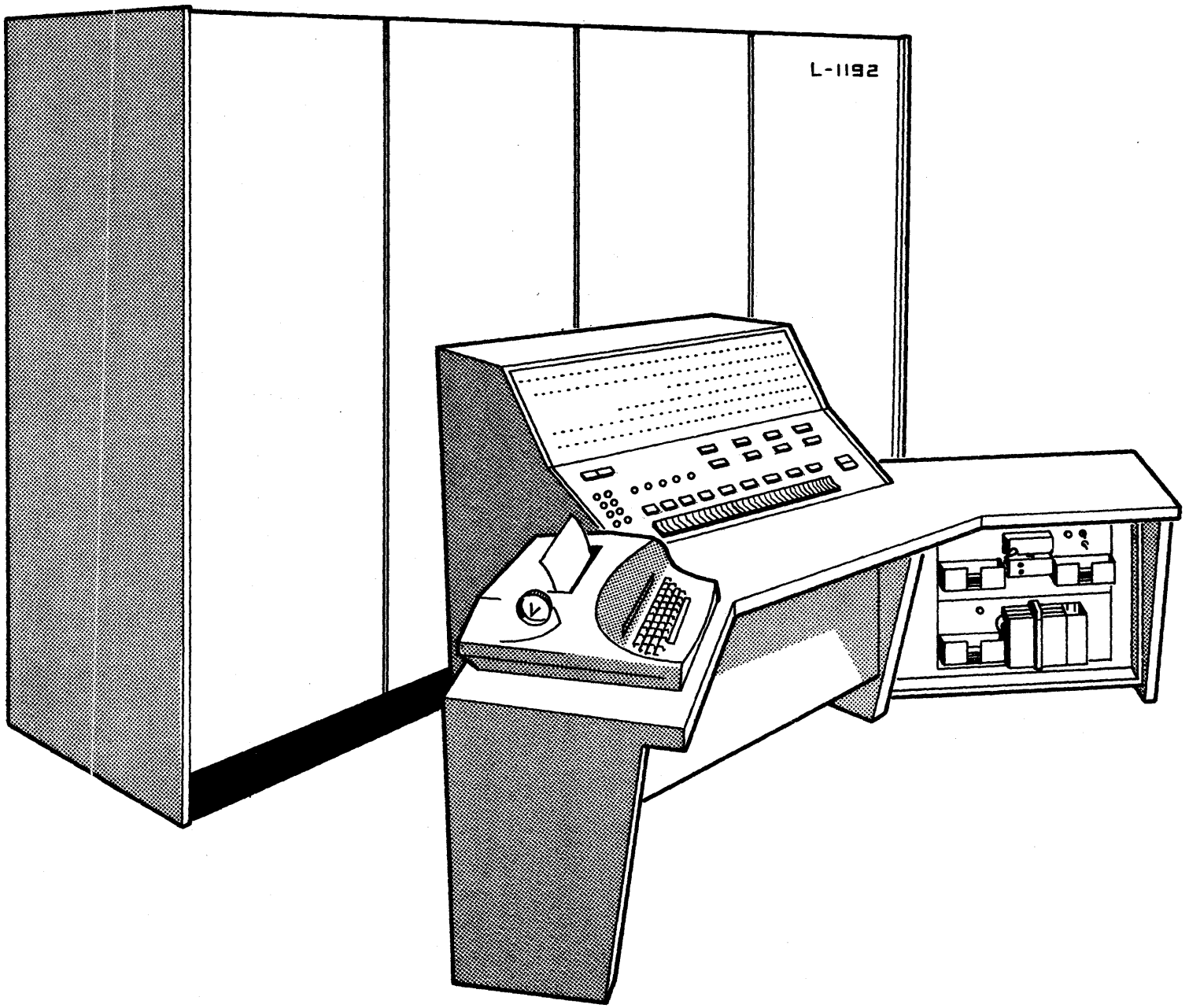


Figure 5-1. L-1192 Controller-Processor

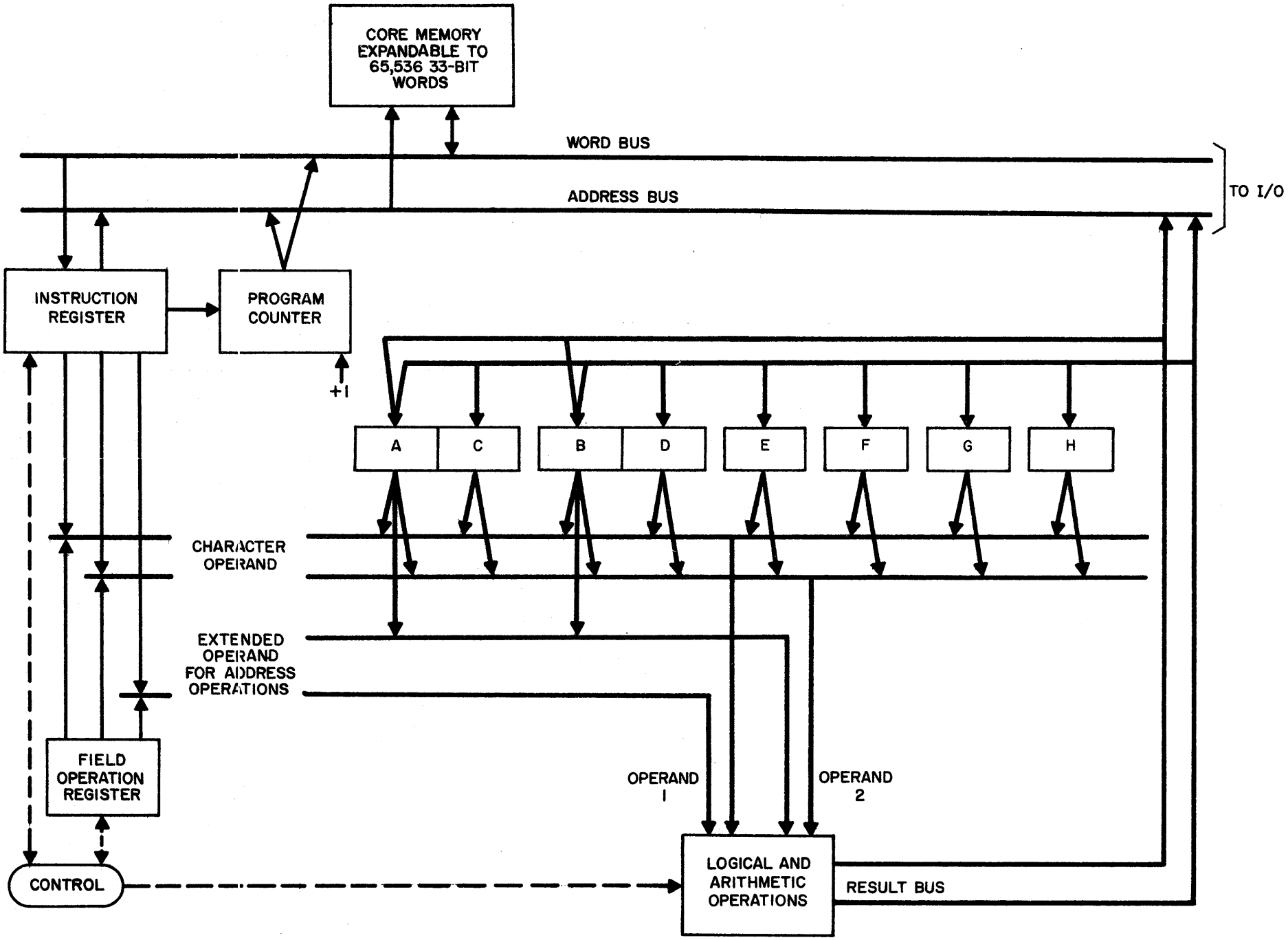
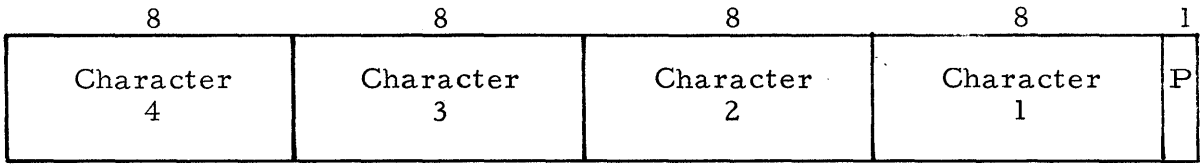


Figure 5-2. Arithmetic and Logical Unit

The simple yet flexible organization of the L-1192 is apparent from the block diagram. Any operations which involve two operands can be obtained in any combination from the data registers in memory by general bus logic. The functions of address modification and incrementation of the instruction counter (P Counter) are also performed by this technique. Communication with the core memory is by an address and word bus which is time shared with the Communications Interface and with the I/O Interface. The relationship between the registers is shown in Figure 5-2.

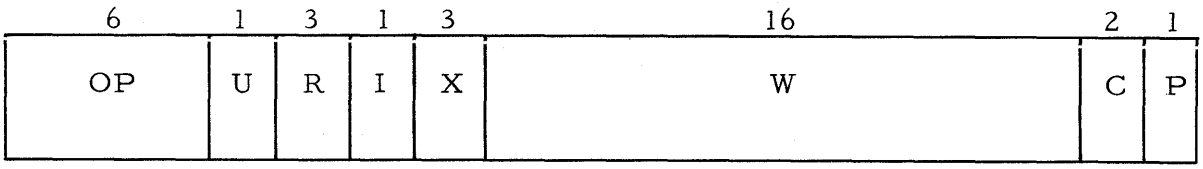
5.3.1.1 Data Format. Each L-1192 word consists of 32 data bits plus one parity bit. Four 8-bit data characters are contained in a word. These characters are individually addressable by a L-1192 instruction.

Data Format



5.3.1.2 Instruction Format. Instructions are one word in length and permit data register selection, in addition to memory addressing. Special features of the instruction set include direct character selection, homogenous word and character indexing, indirect addressing and the inclusion of specialized instructions. The general instruction format is presented below. A complete list and description of the instruction set is provided in Appendix B.

General Instruction Format



Field	Length (bits)	Description
OP	6	Specifies the operation code, permitting up to 64 different instruction types.
U	1	Modifies command execution, as described for individual instructions.
R	3	Specifies one of eight character-length data registers (A through H) containing either the first operand or the operand address.
I	1	Indirect address flag. If I = 0, the W/C address is used directly as the second operand address. If I = 1, W is used as an address from which to obtain the operand to be used. Note 1: If X = 2 through 7, and I = 1, the indexing operation precedes the indirect addressing. Note 2: Multiple levels of indirect addressing and indexing are permitted, as determined by the I and X fields in the accessed word.
X	3	Specifies one of the six index registers. The indexing operation adds the content of the index register to the W/C address field prior to its use. If X = 0, no indexing takes place. If X = 1, the least significant bits of W/C are taken as a direct operand.
W	16	Specifies one of 65,536 word locations in core memory, and provides the second operand address. If W and C are both zero, the operand itself is contained in the data register specified by X.
C	2	Specifies one of four characters within the word for character operations. C may usually be considered a homogeneous address in conjunction with W to provide a W/C character address.

5.3.1.3 Memory Addressing. The address field of an instruction provides for the direct reference to any one of 65,536 words of core memory. In addition, two bits appended to the low order end of the address field allow for the selection of an individual character within the word. This feature permits full character addressing.

5.3.1.4 Indexing and Indirect Addressing. The eight data registers may also function as index registers. When used as index registers, four of the data registers operate independently. Each can provide a total modification of 256 character or 64 word addresses. The remaining four are paired to provide two index registers, each with an addressing capacity of 65,536 characters. Indirect addressing is also a feature of the L-1192. This indirect addressing may extend over any number of levels, depending upon the setting of the I bit in each indirect level. Indexing is effective at each indirect level. This feature permits an instruction effective address to be modified by any number of index registers, as desired.

5.3.1.5 Error Checks. To insure message protection, the L-1192 employs several internal error checks. These include a test of word parity when a word is transferred to or from core memory, an indication if an attempt is made to execute an illegal instruction, and program testing of the parity of characters received from the communication lines. Error checks are also performed on transfer operations between the L-1192 and the two IBM Disk Files or with the IBM 1460 computer.

5.3.1.6 Interrupts. To permit efficient servicing of the Input/Output and Communication Interfaces, a number of interrupts have been provided. (See Figure 5.3-3). The conditions which can cause an interrupt include: Interval Timer elapsed, I/O operation ended, and service requests from either the 1301 or the 1302 Disk Files, as well as from the 1460.

The Interval Timer, which may be loaded by the L-1192 program, provides a 20-bit counter with a maximum time capacity of about 4.5 hours. It will count down at a rate of 1 MC, and generate an interrupt when the counter reaches zero. The Interval Timer may be used to insure that the

Communication Interface will be serviced on a timely basis, or for timing any other required event.

When an interrupt occurs, the content of the P counter and a bit indicating the cause of the interrupt are stored in core memory location ∅, and the next instruction is taken from location 1. All other interrupts are then prevented from occurring. They may be enabled by use of the ESI or RSI instructions. A detailed diagram of the interrupt logic is shown in Figure 5-3.

5.3.2 Communication Interface

The Communication Interface is contained in the L-1192 Controller-Processor Console, and transfers data between core memory and on-line communication channels or devices. (See Figure 5-4.) It can accommodate up to 128 input and 128 output communication line interface modules, contained in up to 16 Line Group modules housed in two Line Unit Consoles. The Communication Interface can also accommodate up to 16 directly connected input-output devices, such as electric typewriter and paper-tape equipment.

5.3.2.1 Core Buffering. Only character buffering is provided by the interface modules, while message blocks are accumulated in core memory. This approach results in significant saving in system cost, and permits the program to operate asynchronously with line rates.

The core buffering operation is performed in a unique and efficient manner. Two cyclic table areas, named Channel Activity Tables, are reserved in core memory. These are automatically loaded by the Communication Interface with the input characters or with requests for output characters, each with a line identity number. Channel Activity Table 1 (CAT 1) is reserved for character requests made by high speed output lines. Channel Activity Table 2 (CAT 2) is used to buffer all other lower priority output requests and all input traffic. CAT 1 may contain 128 entries or words in memory, while CAT 2 has a capacity of 256 entries.

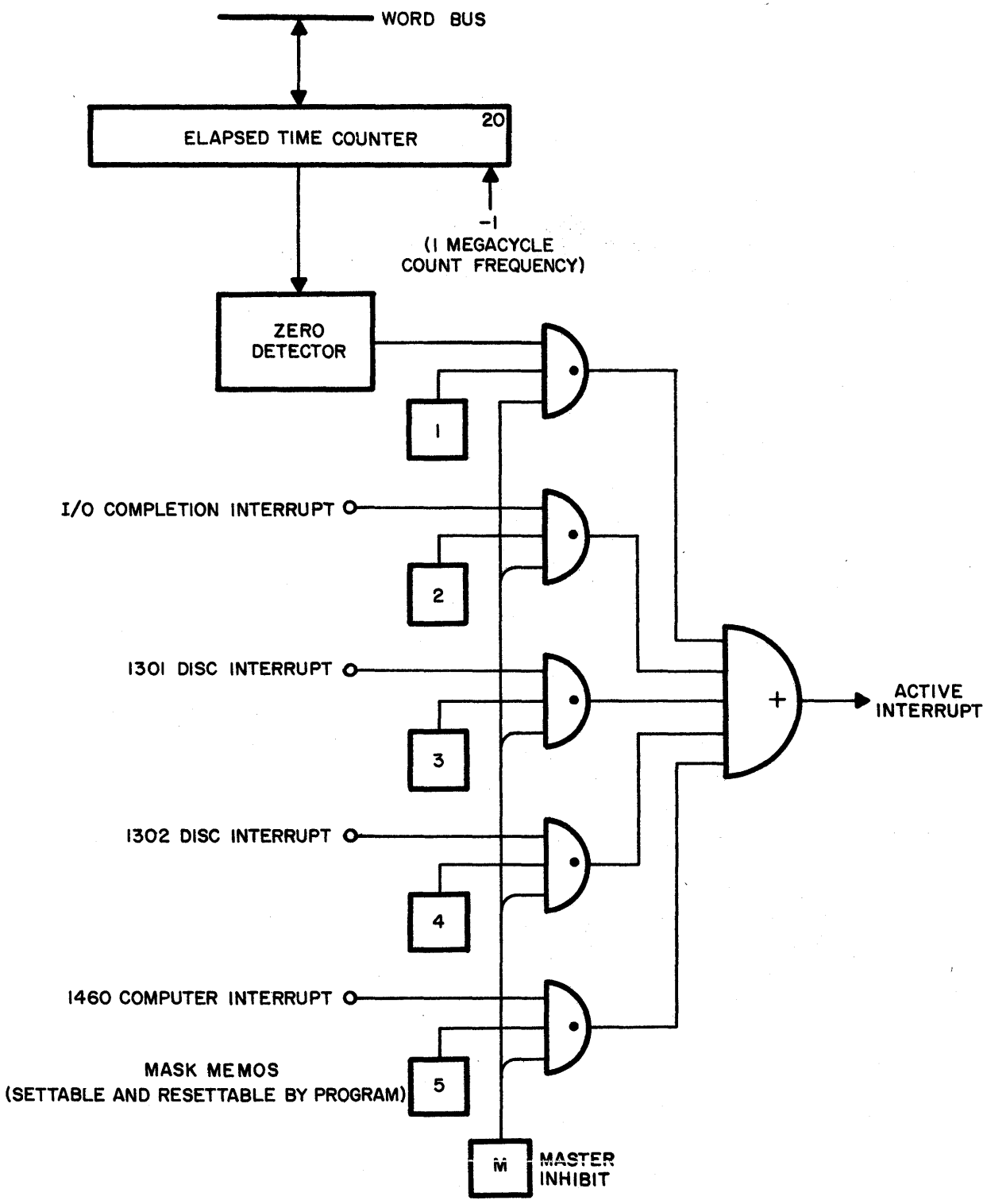


Figure 5-3. Interrupt Logic

5.3.2.2 Communication Line Servicing. The Communication Interface makes use of two cyclic load address registers designated L1 and L2, for CAT 1 and CAT 2 control respectively. Each time an entry is to be stored in either table, the appropriate load counter provides the required core loading address and is then stepped. CAT 1 and CAT 2 are unloaded and processed by means of two other cyclic address counters, named U1 and U2. A more detailed discussion of the procedure for unloading of CAT 1 and CAT 2 is contained in the discussion of the Initiate Character Cycle instruction found in Appendix B. All load and unload counters are reset by clearing the L-1192, or by depressing LOAD PAPER TAPE on the Operator's Console.

The output characters transmitted in response to output requests are automatically obtained from another table, named the Line Character Buffer Table. The LCBT contains one assigned character address in core memory for each output line implemented. These output locations are loaded by the program with the next output character for the corresponding line number. When an output interface module has completely transmitted an output character it initiates an interface available signal, which will be detected by and halt the scanners in both the Line Group module and in the Communication Interface. The two scanner positions will generate the proper character memory address in the LCBT, and the character at this position will transfer to the waiting output line interface module. The Communication Interface will then load an output request, including the line identification number, into CAT 2 (for high speed lines) on into CAT 1 (for slow speed lines). The L1 or L2 load point is stepped, and the scanners are released to continue searching for other input or output traffic. Input traffic is handled in a somewhat similar manner by the Communication Interface. When a complete character has been accumulated by an input line interface module, it sets a signal causing the scanners to stop. Memory access is obtained, and the character plus the associated line number is automatically transferred to the next available word location in CAT 2. The L2 load point is stepped, and the scanners are then released to continue looking for traffic. The input character will be processed by the L-1192 program, whose instructions can efficiently

access the next input character or output request from the two Channel Activity Tables.

5.3.2.3 Low Speed I/O Servicing. Low speed character-serial bit-parallel devices are also serviced by means of the Communication Interface. Input and output character transfers are similar to those for the communication lines. The devices include such equipment as typewriters, paper tape readers, and paper tape punches. Up to sixteen (16) duplex channels, in addition to the communication channels, are provided for by the Communication Interface. Of these, one duplex channel will be utilized by the console typewriter, and a second shared by the L-1192 paper tape reader and paper tape punch. Fourteen channels will therefore still be available for attachment to additional customer equipment.

The characters transferred between the L-1192 and these devices are processed exactly the same as characters from the communication lines. Scanner address Ø is reserved for the paper tape reader and paper tape punch. Address 20_8 is reserved for the typewriter. Transfers to or from these devices are character-serial, with up to 8 bits/char., and input parity may be checked by the L-1192 program.

5.3.2.4 Channel Activity Table Formats. The following information provides the word formats loaded into the Channel Activity Tables (CAT) for the input characters and the output requests, and indicates the available identification addresses that may be assigned to the communication lines and to the I/O devices:

	8	8		8	8
Input Entry	Not Used		0	Line/Device Address	Input Char.
Output Request	Not Used		1	Line/Device Address	Always Zero

Line/Device Address (Octal)	Device
∅	L-1192 Console Paper Tape
20	L-1192 Typewriter
40-360	Optional Low Speed I/O
400-477	Line Interfaces in Line Unit Console #1
500-577	Line Interfaces in Line Unit Console #2 (Optional)

Note: For device addresses below 400, the least significant 4 bits are always zero.

5.3.3 I/O Interface

The I/O Interface provides for data transfers to and from the IBM 1301 Disk File, the IBM 1302 Disk File (shared with the IBM 7090), and the IBM 1460 computer. All data transfers to and from these devices occur on an asynchronous basis with the I/O interface obtaining memory access priority as needed. The I/O interface will independently execute interface block transfers, and overlap central controller operations. The block diagram of the I/O interface is shown in Figure 5-5.

5.3.3.1 Description. The I/O interface contains a Device Command Register, a Block Control Register, an Interface Data Register, a Command Address Register, and associated control and bus logic. Once the interface has been placed in operation, it will access a device command word from core memory, which may demand one or more Block Control words. Data is shifted into or out of the Interface Data Register in a character-serial fashion on a device demand basis. As data is transferred, the Block Control Register counts characters and memory locations.

When each Block Control word is satisfied, the status of the disconnect bit is tested. Depending on its state, the operation is terminated, or the

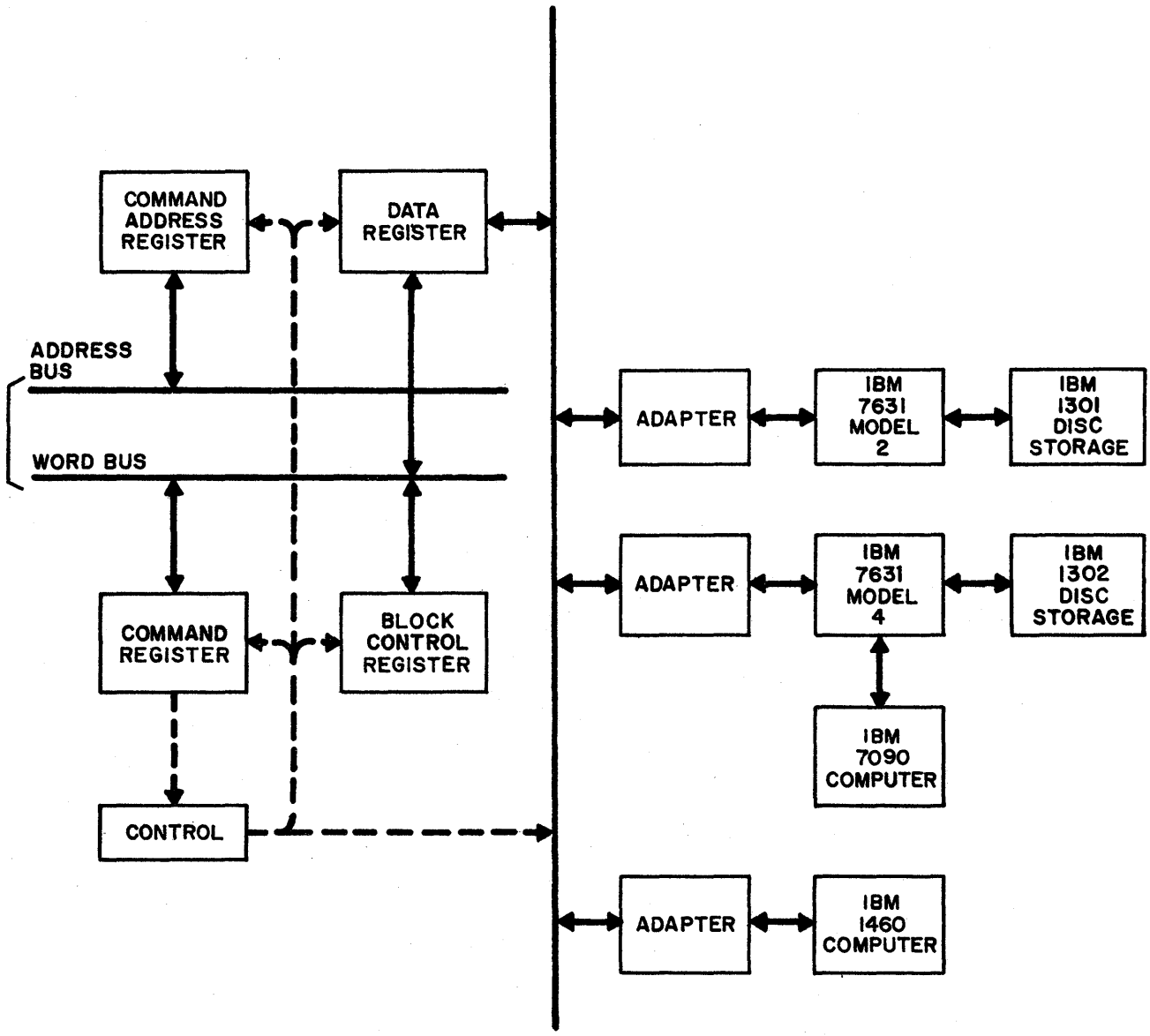


Figure 5-5. Input/Output Interface

next Block Control word is accessed and the transfer continues. When the transfer terminates, the main processor program will be interrupted (see section 5.3.1.6).

5.3.3.2 Operation. The I/O Interface is initially placed in operation by an Initiate I/O Operation (IIO) instruction. This instruction will cause its address (W field), after modification by indexing and indirect addressing, to be transferred to the Command Address Register (ICA) in the interface. The IIO instruction will terminate, and the interface will immediately demand memory access. It will fetch the word at the location specified by ICA, which must be a device command as indicated by a one in bit 32. The device command will be loaded into the Command Register (CR) and decoded. Bit 31 determines whether the command requires the movement of data. Bits 27-30 select the device. Bits 17-26 specify the actual device operation.

The device select field of the command gates the control and data lines to the desired device. The device operation field is decoded partially in the CR and partly in the device adapter to provide the necessary control of data transfer in the interface and to actuate the particular control lines within the device.

If Bit 31 of the command were zero, the interface immediately goes Not Busy and is free to execute another operation at the request of the L-1192 main program.

If Bit 31 is a one, this command will require the transmission of data. The word following the interface command in core memory is obtained by means of ICA. Bit 32 of this word must be zero to denote a Block Control word. The Block Control word is placed in the Block Control Register. Depending on the device operation, data is transferred to or from the Interface Data Register. If this is an output operation, a word is accessed from core memory before the transmission is started, by means of the address contained in the Block Control Register.

As data is transferred, the count and address of the Block Control Register are stepped. When the character count reaches zero, Bit 31 of

the Block Control Register will be checked. If a one, the next Block Control word will be accessed using ICA. If this bit is zero, the operation will be terminated and the main processor program interrupted.

5.3.4 Operator Console

An Operator/Maintenance console is provided with the L-1192 Controller-Processor, and contains all controls, indicators, and devices necessary to operate the L-1192 Controller-Processor. It has been designed to facilitate system operation, system fault location, and program checkout.

The Operator Console is shown in Figure 5-1. It consists of three sections. The left-hand section contains the console typewriter. The right-hand section contains the paper tape punch and the paper tape reader; the center section (Figure 5-6) contains the controls, indicators, and displays necessary to operate the system.

5.3.4.1 Displays. All of the important registers are separately displayed by means of neon indicators. These lamps are also used to display the contents of selected locations in core memory. The Registers displayed include: the 8 Data Registers, the Program Counter, the Field Operand Address Register, the Instruction Register, and the Memory Register. The operator panel also contains indicators which provide the status of internal errors, and the condition of the various input/output devices.

5.3.4.2 Data Entry. The Program Counter display contains integral push switches to permit the selective setting of any bit position. The remaining registers and core memory are set by means of 33 data entry keys, which are located toward the bottom of the Control Panel. These permit a full word of data to be set up and entered into the Instruction Register, either group of Data Registers, or the Memory Register. The Data Register may be loaded from or stored into core memory by means of the appropriate panel switches. The address accessed in core memory is set into the right-most data entry keys.

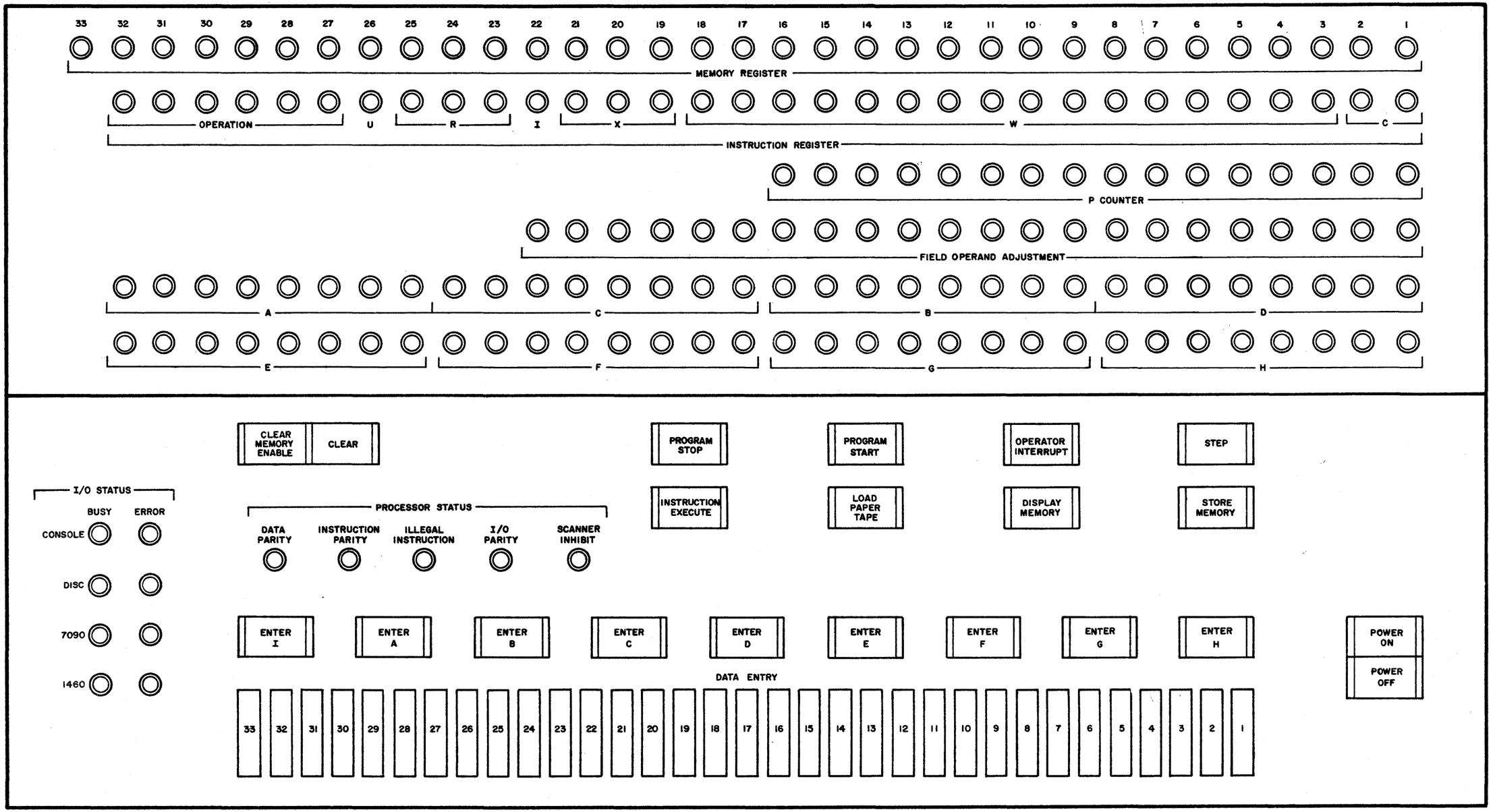


Figure 5-6. Operator Console Control Panel

5.3.4.3 Controls

1. Clear Functions

The registers alone or the registers and memory together may be cleared. Depression of CLEAR will clear only the registers in memory and set the ALU to zero. The data contained in core memory will be left unchanged. However, if CLEAR MEMORY ENABLE is depressed simultaneously with CLEAR, the data contained in core memory will be cleared as well. This feature provides a rapid clearing of the data in memory without the aid of a program, while at the same time providing a degree of protection against accidental clearing by an operator.

2. Run Control

The "rest" state of the L-1192 is the Halt mode. When in the halt mode, the PROGRAM STOP button will be illuminated red. Depression of PROGRAM START will cause the computer to enter the Run mode and begin instruction execution at the location specified by the P Counter. While in the Run mode, the PROGRAM START button will be illuminated green, and the PROGRAM STOP button will be extinguished. Subsequently, depressing PROGRAM STOP, executing a halt command, or clearing will cause the system to return to the Halt mode.

While in the Halt mode, single instructions may be executed by depressing the STEP button. Each time this button is depressed the instruction contained in the Instruction Register will be executed, the next instruction will be fetched from the location specified by the P Counter, and the P Counter stepped by 1.

3. Load Paper Tape

Programs can be automatically loaded from the paper tape reader. Depression of the LOAD PAPER TAPE button will cause a built-in sequence to be executed. This sequence will cause 64 words (256 characters) to be loaded from the paper tape reader into the CAT 2 buffer table, and will execute a jump to the start of that table. The instructions loaded in this manner may then be used to load programs into any desired memory area. The loading process is described in detail in Section 5.3.4.5.

4. Manual Controls

Controls have been provided for executing individual instructions, and displaying and modifying memory. The INSTRUCTION EXECUTE button will cause the content of the Instruction Register to be executed. The next instruction will not be executed, and the P Counter will not be stepped. Data may be displayed from memory by depressing the DISPLAY Memory button. The word at the location specified by the right-most data entry keys will be displayed in the Memory Register. Depressing the STORE MEMORY button will cause the word presently in the Memory Register to be stored at the location specified by the right-most data entry keys.

5.3.4.4 Input/Output Devices

1. Typewriter

The L-1192 Typewriter provides both an input and output function. It operates via the Communication Interface, and utilizes a low speed I/O position. The Typewriter prints at 15 characters per second, and is located at the Operator Console.

2. Paper Tape Reader

The L-1192 Photo Reader inputs 8-level paper tape at 300 characters per second. It operates via the Communication Interface, and utilizes the input half of a low speed I/O position. The Photo Reader is located at the Operator Console.

3. Paper Tape Punch

The L-1192 Paper Tape Punch outputs 8-level paper tape at 60 characters per second. It operates via the Communication Interface, and utilizes the output half of a low speed I/O position. The Paper Tape Punch is located at the Operator Console.

5.3.4.5 Program Loading. The normal method of initially loading programs into the system is by means of the L-1192 paper tape reader located on the operator console. An automatic load sequence is provided,

which is initiated by depressing the LOAD PAPER TAPE key on the Operator Console. This inputs a total of 256 characters into CAT 2, in a packed format, and initiates program execution at the starting location of CAT 2. The first data loaded into memory should consist of a self-relocating loading program, which will relocate itself out of CAT 2 and then call in the remaining instructions to be loaded. For system operation, programs should be retained on and loaded from the 1301 Disk File.

When LOAD PAPER TAPE is depressed, the following operations occur:

1. The load function is locked, and L2 (for CAT 2) is reset to zero.
2. The scanner is locked at the L-1192 paper tape channel (address Ø) and the paper tape reader is started. Logic is enabled to cause L2 to count character addresses rather than word positions.
3. 8-bit characters are loaded into CAT 2, starting at the least significant character address. The device address is not input.
4. When 256 characters have been loaded in CAT 2, the Program Counter is set to the starting location of CAT 2, the load function is released, and the L-1192 is placed in the RUN mode.

The load program entered into CAT 2 must consist of 64 words in packed format, containing one word (32-bits) of instruction or data for each four 8-bit characters on tape. It must relocate itself outside CAT 2 before it calls in additional information from the paper tape reader. The program and data characters input and processed by the loader routine may then contain parity check information.

5.3.5 I/O Adapters

A total of three I/O Adapters are provided with the proposed system, and are contained in the L-1192 Controller-Processor Console. There are two types of Adapters utilized.

5.3.5.1 L-1192/IBM 7631 Adapter. The IBM 7631 File Control Unit is a controller for either the 1301 or the 1302 Disk File. An adapter that will operate with the different models of the 7631 is provided between the 1192 I/O Interface and the 7632 Model 2 controlling the 1301 Disk File. Another such adapter is utilized for the interface with the 7631 Model 4, providing shared 1192/7090 access with the 1302 Disk File. The adapter has two principal functions: (1) to perform signal-level adjustments, and (2) to convert standard L-1192 control signals into the specific controls required by the 7631. Data formats are consistent between the 7631 and the I/O Interface, requiring no format changes.

5.3.5.2 L-1192/IBM 1460 Adapter. The IBM 1460 Data Processing System communicates with the L-1192 by means of the IBM Serial Input/ Output Adapter. The L-1192 I/O Interface provides an adapter to produce appropriate signal levels and control functions. Certain characteristics of data exchange between the 1460 system and the L-1192 are further determined by jumper-wiring of the Interface Plugging Lines. The Interface Plugging Lines are located in the 1441 Processing Unit of the 1460 system.

5.4 PERIPHERAL EQUIPMENT

5.4.1 Random Access

5.4.1.1 1301 Disk File. The random-access memory proposed for use with the L-1192 system is the IBM 1301 Disk File, used in conjunction with an IBM 7631 Model 2 File Control Unit. This system provides for storage of 21.60 million 8-bit characters or 27.96 million 6-bit characters. The average time required to read records from this disk is approximately 180 μ sec per record. For writing and verifying, the average time per record is approximately 220 μ sec.

The storage media itself consists of 40 surfaces coated with a magnetic recording material. Each surface has 250 tracks, any one of which may be accessed by a moving head assembly. In a given position of the head assembly, 40 tracks (one from each surface) are available for access

through electronic switching of the heads. The disks spin at a maximum rate of 1790 RPM.

5.4.1.2 7631 File Control Unit. The IBM 7631 Model 2 File Control Unit now available to the IBM 7090 system at Sandia will be utilized to interface between the L-1191 and its IBM 1301 Disk File. In the L-1192 system the 7631 performs the following functions:

1. Transfers addresses between the L-1192 and the disk.
2. Interrupts the L-1192 when the disk has completed a head positioning (SEEK) operation.
3. Searches the file tracks for a particular record addressed by the L-1192.
4. Obtains data on a character-by-character basis from the L-1192 I/O Interface, and transfers it to the disk in a bit-by-bit manner.
5. Accepts data on a bit-by-bit basis from the disk and transfers it to the L-1192 I/O Interface on a character-by-character basis.
6. Provides status information to the L-1192 on demand.

The 7631 File Control will be connected to the L-1192 through the use of a simple adapter unit. This unit will serve to translate control or operation codes decoded by the I/O Interface into logic signals acceptable to the 7631.

5.4.2 IBM 7090 Interface

The method proposed for interfacing the L-1192 with the IBM 7090 computer is through shared access to its present 1302 Disk File. This approach requires the replacement of the current 7631 Model 2 with a 7631 Model 4 File Control Unit. This method of interfacing has several advantages:

1. The 7090 will be attached to a piece of compatible equipment, thereby greatly reducing changes to existing programs.
2. The 7631 Model 2 currently used by Sandia on the 7090 can be used by the L-1192 to access a 1301 Disk File. This minimizes the amount of new equipment that will have to be acquired.

3. Due to the similarity of the Model 2 and Model 4 File Control Units, a majority of the disk I/O circuitry in the L-1192 can be shared between the two devices, thus effecting a significant saving in machine cost.

4. The 7631 Model 4 can be used by the L-1192 during acceptance testing at the Librascope facility. This procedure will remove the need for obtaining additional equipment on a temporary basis to accomplish system checkout.

5.4.3 IBM 1460 Interface

The interface between the L-1192 I/O Interface and the IBM 1460 will be accomplished via the Serial I/O Adapter (IBM special feature No. 7080) which has already been ordered with the 1460 for Sandia.

Data transfers through this interface will be character-serial, with the least significant character of the lowest addressed word of the block being the first to be transmitted. Through the use of interface plugging lines associated with the Serial I/O Adapter, it is possible to wire the interface to be completely compatible with L-1192. The L-1192 will be capable of providing signals which the 1460 can test by means of its program. The 1460 will be able to interrupt the L-1192 processor program to demand service. In this way the two systems will be able to work together at maximum efficiency.

Section 6

PROGRAMMING TECHNIQUES

In order to realize the processing speeds for which the L-1192 Controller-Processor was designed, efficient programming techniques should be followed. This is particularly true of the character processing cycle, which for the Sandia application may be executed as frequently as 10,000 times per minute. Since this cycle is so short, savings of only a few instructions will result in significant reductions in processing time. This section contains some suggested procedures for servicing the input-output characters and providing the message storage, retrieval, and processing.

6.1 COMMUNICATION INTERFACE SERVICING

The operation of the communication interface hardware is described in section 5.3.2. The communication interface service routine must perform the following functions:

Obtain from CAT2 all input characters. These must then be translated to a common internal code and compiled into message blocks.

Obtain requests for output characters from CAT1. Upon recognition of an output character request, the communication interface service routine must obtain the next character to be output, translate it, and store it in the LCBT.

6.2 I/O INTERFACE SERVICING

The I/O interface operates asynchronously with the processor program. Its detailed operating description is contained in section 5.3.3. To realize the greatest efficiency in I/O usage, requests for data transfer to and from the various I/O devices should be executed by a single service routine. Any other routine will then store a request for data transmission in a communication table. Based on device priorities, the I/O service

routine will then attempt to execute these requests. When a request is successfully executed, the I/O service routine will then update the bookkeeping markers associated with the table involved in the data transfer. The I/O service routine will be entered on all I/O interrupts. When an I/O interrupt occurs, its general cause may be determined by accessing the content of location Ø, which will show whether the interrupt was caused by a device service request or an I/O end of data transfer signal. To determine the reason for completion, the I/O registers and conditions should be returned to core memory and tested.

Service requests will be generated by the disk memories at the completion of a SEEK operation. While this operation was in progress, other I/O operations were permitted to take place, and may be in progress at the time the service request is received. However, the actual disk data transfer which required the SEEK should be initiated as the next operation.

Precautions must also be taken to detect an I/O "Hang-up". If through some equipment or program malfunction, an I/O operation is initiated which can never terminate, the operation must be forcibly halted by the program and a program indicator set to prevent its use and to alert the operator. It is suggested that this hangup condition be detected by counting the number of interval timer interrupts which have occurred since the last I/O interrupt.

6.3 MESSAGE PROCESSING

Messages passing between the character processing routine and the main processing and editing routines should be buffered on disk to allow operation under sustained high character rates. It is recommended that messages be broken into blocks of approximately 100 characters each. Each input and each output line will then require a 100 character buffer area to contain the block currently being received or sent. Through the use of dynamic memory allocation, a small reserve of core memory space can be kept to take care of message blocks awaiting transfer to or from the disk.

A message queue will be maintained by the character processing and main cycle routines such that, when a message is received from a line, it will be written on disk, and all disk locations of the blocks of that message will be placed at the bottom of the queue. The main cycle will read the top of the queue, and will request the message blocks from the disk as they are needed for processing. When the message finally leaves the L-1192 processor and enters the 7090 and 1460, and this transfer has been verified, it can be deleted by placing all of its block addresses in a list of available disk storage spaces.

Similarly, when a message enters the L-1192 system for transmission out on one or more lines, the disk locations of the message blocks will be placed at the bottom of the output queue for each line over which it is to be sent. As it is sent on each line, a count unique to the message will be stepped. When the message has been sent to all addresses, and when the time has elapsed for retaining the message, it will be deleted from the disk by placing its block locations in the disk availability list.

For protection of messages, in the event of computer failure, the disk message location table and the disk availability table should be periodically written on the disk.

6.4 OPERATIONAL PROGRAM STRUCTURE

This section discusses the operational program for a typical communication controller, and how it might be used to carry out the communication processing requirements of the Sandia system. The overall flow chart of this program is presented in Figure 6-1.

6.4.1 Interrupt Cycle

The interrupt cycle is entered as a result of any system interrupt. Immediately upon entry, the interrupt cycle must determine the cause of the interrupt. If the interrupt resulted from an I/O completion or service request, the I/O routine will be entered before character processing is initiated. If the interrupt occurred as a result of the interval timer elapsing, the character processing routine must be entered directly.

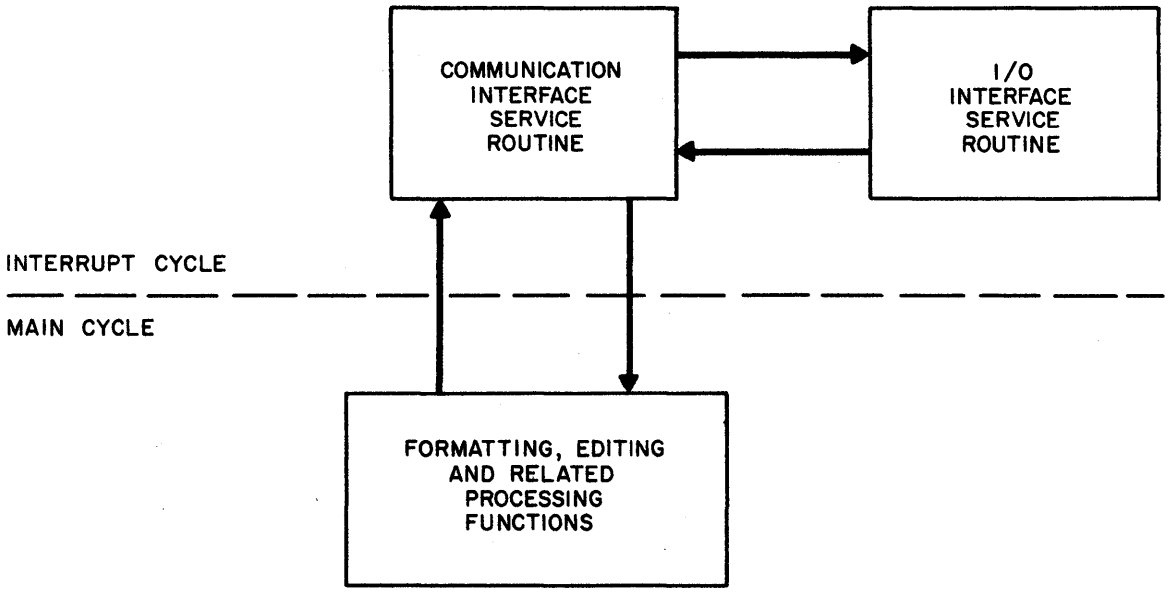


Figure 6-1. Flow Chart - Overall Operation Program

6.4.1.1 I/O Service Routine. The I/O routine will determine the specific cause of the interrupt. A general flow chart for this program is presented in Figure 6-2. If caused by an abnormal end of transmission, such as a data error, the operation must be re-initiated or the device set unavailable to the program and the operator notified.

If the interrupt resulted from normal I/O completion, the bookkeeping markers for the table affected by the transfer must be updated and the next highest priority operation initiated. The location of the bookkeeping markers to be changed will be obtained by means of a linkage address, which formed a part of the original I/O request.

If the interrupt occurred as a result of a disk attention request, it indicates a previously initiated "SEEK" operation has been successfully completed, and the requested data may now be transferred. If the service request originated in the 1460 computer, the I/O routine will have to initiate a previously requested transfer.

When the next highest priority operation has been initiated, the I/O routine must exit to the character processing routine.

6.4.1.2 Character Processing (Communication Service) Routine. This routine is entered directly on an interval timer interrupt, or after I/O processing on any other interrupt. It has the function of answering all requests for output characters which entered CAT1, and processing all input characters received in CAT2. A general flow chart for this program is presented in Figure 6-3.

CAT1 is processed first, since the timing of output characters is more demanding than input. For each character request, the message block storage area corresponding to that time is accessed from core memory and the next output character obtained. This character is translated to the desired line code by means of indexed table look-up, and stored in the character position corresponding to that line in the Line Character Buffer Table.

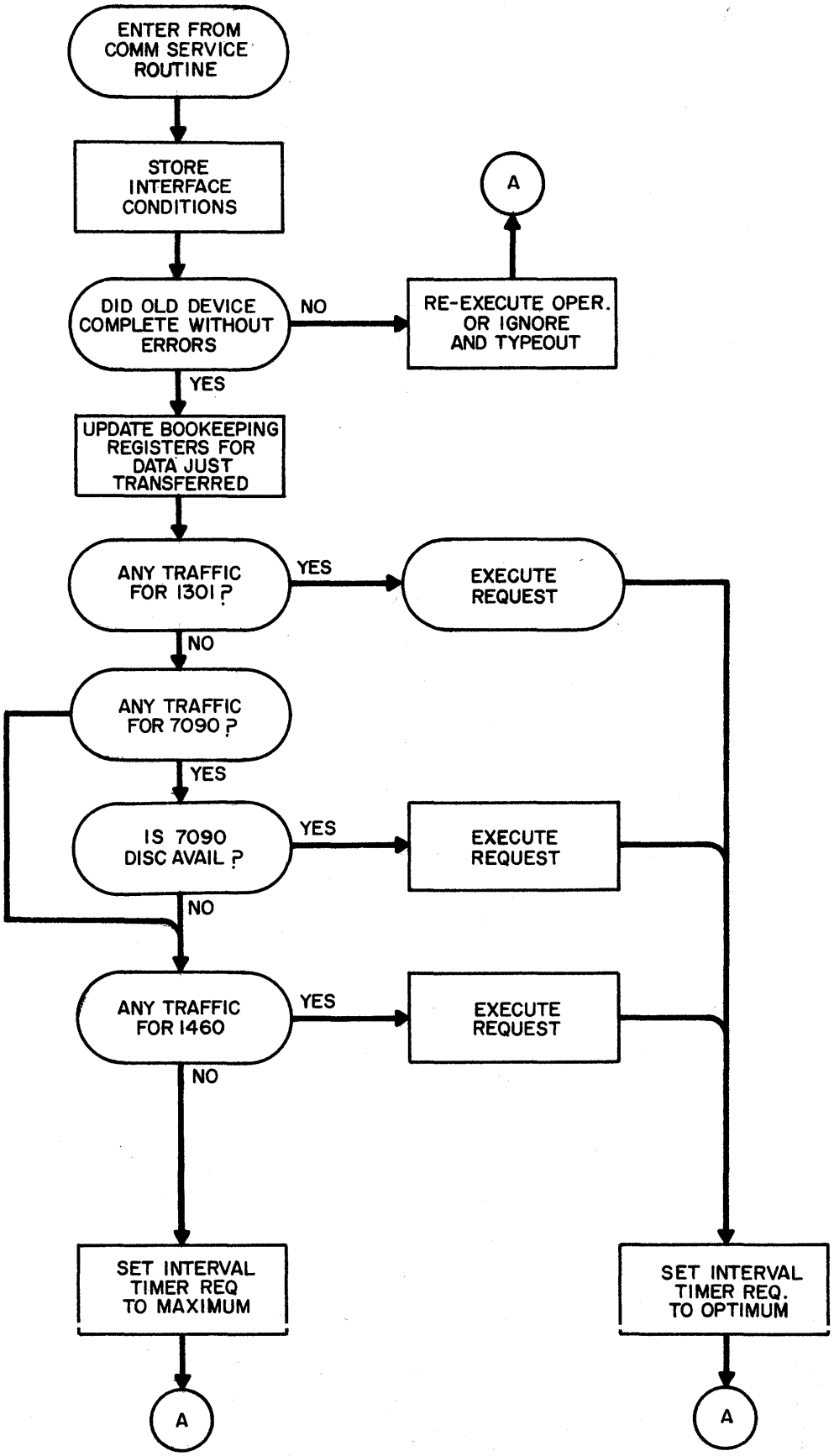


Figure 6-2. Flow Chart - I/O Service Routine

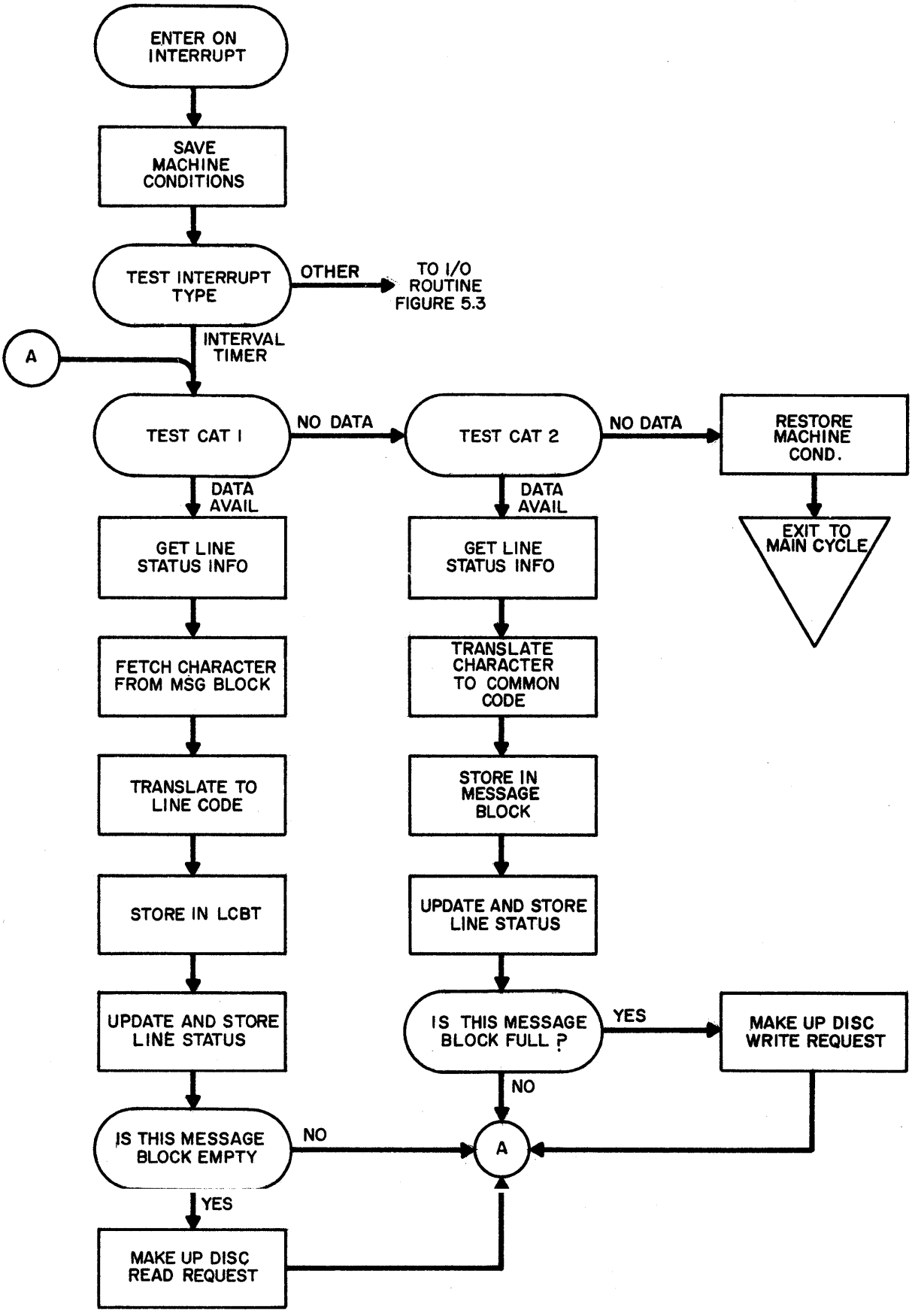


Figure 6-3. Flow Chart - Communication Service Routine

When all entries in CAT1 have been processed, the input characters which have been buffered in CAT2 will be processed. Each character is obtained from CAT2, translated to common code, and stored in the appropriate message block buffering area.

When both CAT1 and CAT2 are empty, the machine conditions are restored, the interrupts are re-enabled, and the processor exits back to the main cycle.

6.4.2 Main Cycle

The exact functions of the Main Cycle are dependent on the complete application requirements of the Sandia Corporation. However, certain assumptions can be made, and a programming method can be recommended based on these assumptions. The most important main cycle function will be the routing of message blocks between the central controller and the 1301 disk, the 7079 disk and the 1460 computer. Additional functions will include changes to message formats to resolve incompatabilities between the various line devices and the 7090 and/or 1460, and various editing and data checking functions to insure message validity. A message processing routine is therefore recommended which will function as a sort of executive routine. It will take care of such matters as message routing within the communication controller environment, the scheduling of out-going messages to the lines, the control of format conversion, the control of editing, and the requesting of routines to be read from disk program storage. Figure 6-4 depicts a typical flow chart for such a main cycle layout. Using this method, there would be a group of relocatable routines which would be highly specialized in performing some formatting or editing function. These would be called in from the 1301 disk as needed, based on the type of message being processed. The executive routine would maintain control over a dynamic program core storage area, where these routines would be kept while being executed.

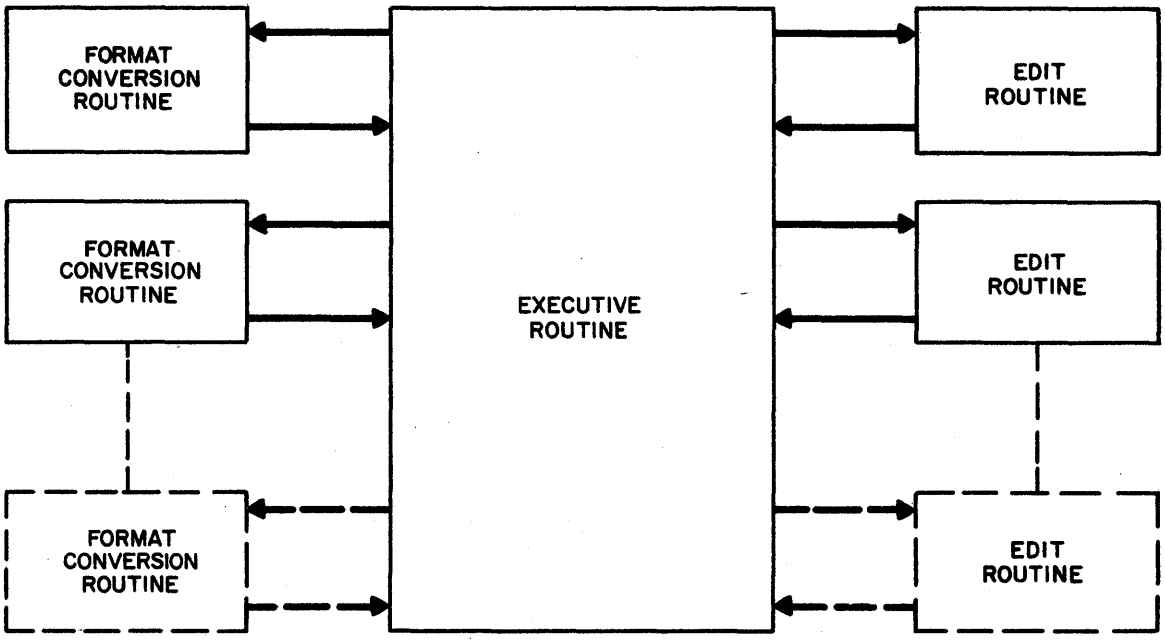


Figure 6-4. Flow Chart - Main Cycle Organization

Section 7

SUPPORT SOFTWARE

This support software offered with the L-1192 system will include an Assembler (LAP-1192), an I/O Control Routine, a Multiply Subroutine, a System Loader, a Core and Disc Dump, and Acceptance Tests. Brief descriptions of these programs are included in the following paragraphs. Detailed descriptions of the Assembler and the I/O Control Routine are in Appendix C.

7.1 LAP-1192

This program will accept symbolic instructions of the appropriate format punched into paper tape. It will generate a paper tape in absolute machine language, suitable for later loading via the paper tape reader. It will also provide a typewritten listing of the input symbolic instructions and the resultant machine code. As an additional option if required, Librascope will modify LAP-1192 and provide a corresponding IBM 1460 program so that the symbolic instructions may be read in from punched cards or magnetic tape via the IBM 1460, and the output listing printed by the 1460.

7.2 I/O CONTROL

The I/O Control program maintains control of all data transfers to and from the various I/O devices. It is entered on an I/O interrupt and serves to utilize the disc and other equipment in a highly efficient manner. Requests for data transfer are stored by the processing routines in intercommunication tables for each I/O devices. The I/O routine will then cause the requests for transfer to be executed and, on successful termination, will update indicators associated with the data transfer region.

7.3 MULTIPLY SUBROUTINE

The Multiply Subroutine utilizes a table look-up method to permit the multiplication of two unsigned decimal fields of variable length. It requires about 60 words of core memory for storage of the routine and its associated table. All registers used by the multiply are restored before returning to the calling program.

7.4 SYSTEM LOADER

The system loader will load programs into core memory via the paper tape reader. The loader itself is self-relocating, and will load absolute or relocatable programs.

7.5 CORE AND DISC DUMP

The dump program will dump specified locations of core memory or disc memory out to the console typewriter. If optional 1460 I/O routines, as described for LAP-1192, are procured, then dumping can occur via the 1460 line printer. Data may be dumped as characters or in command format with mnemonics.

7.6 ACCEPTANCE TESTS

Acceptance tests will be written by Librascope to demonstrate satisfactory performance of the L-1192 Controller-Processor System. These tests will prove the execution of instruction and of I/O operations. After customer acceptance, these programs may be utilized as maintainance routines to detect equipment failures, and will reflect the nature of the malfunction. They will not, however, perform fault diagnosis or attempt to indicate the hardware components which have failed.

Appendix A

LINE INTERFACE MODULES

The line interface modules contained in the Line Unit Console provide character buffering between the input-output communication lines and the L-1192 Core Memory. There are four types of interface modules: synchronous input, synchronous output, asynchronous input, and asynchronous output. A simplex communication line requires either an input or an output line interface module, while a half-duplex or a full-duplex line requires an input-output pair of modules. Each line interface module contains either a character assembly (input) or disassembly (output) shift register. These provide the bit to character or character to bit serial transformations and buffering. Transfer between the interface modules and core memory occurs under the control of a Line Group Scanner and the Communication Interface on a cycle stealing basis.

When the Input Line Interface Module for a receiving line has accumulated a character, a Buffer Full signal is generated which is detected by the Line Group Scanner. The Scanner then stops, and generates a request for a core memory cycle which is detected by the Interface Scanner. On receiving core priority, the interface automatically inserts a word into memory, containing the input character and the line identity. The memory address is determined by the L2 Counter, which is then incremented. The L2 Counter sequentially addresses 256 words (Channel Activity Table 2) in core memory. When the end of this table is reached, it starts over in a cyclic fashion. The content of this cyclic table are processed by the L-1192 program.

Output traffic is handled in a slightly different manner. When the character in an Output Line Interface Module has been completely transmitted, it sets a Buffer Empty signal. This stops the Line Group Scanner, and initiates a request for a core memory cycle. On receiving priority, the interface automatically accesses the next character in an output character position reserved for that line. The interface then inserts an output

request and the line identification into a word position in a Channel Activity Table (CAT). The memory address for the request is determined by the L2 Counter for slow speed lines, or by an L1 Counter for high speed lines. The L1 Counter will step and cycle through the addresses for a fixed location 128-word CAT 1 Channel Activity Table in core memory. The content of CAT 1 is available to the L-1192 program, which in processing the request will normally refill the output character position for that line.

A. 1 SYNCHRONOUS INPUT LINE INTERFACE MODULE

The synchronous input interface module, which serves as a character buffer between one input data channel and the L-1192, is presented in Figure A-1. The data terminal bay provides each input or output interface with a data line and a clock line. The input data is received bit-serially at the low-order end of a 9-bit shift register, and accumulated under control of the clock signal. Each interface may operate at a transmission rate of up to 4800 bits per second. The input line interface may at any time receive a reset signal from the L-1192. This program action turns OFF a Character Frame Flip-Flop, and initiates a character frame sequence. The least significant eight bits of the interface shift register are then continually decoded to detect an "Idle" character. When the Idle code is received, the Character Frame Flip-Flop is turned ON, indicating that the interface has framed a character. This causes the shift register to be cleared to zero, with a "one" inserted in the first (least significant) bit position. This bit is shifted along with input data until it reaches the ninth bit position of the register, indicating that eight bits (a character) have again been accumulated. A "one" in the ninth bit position of the shift register and the ON position of the Character Frame Flip-Flop provide a Buffer Full signal. Transfer of the character to L-1191 memory sets the character received line true, causing the shift register to again be cleared to zero and a "one" inserted in the first bit position. This process continues as each data character is collected in the shift register. The L-1192 program performs the task of counting consecutive framing characters to determine a character frame state. Should the

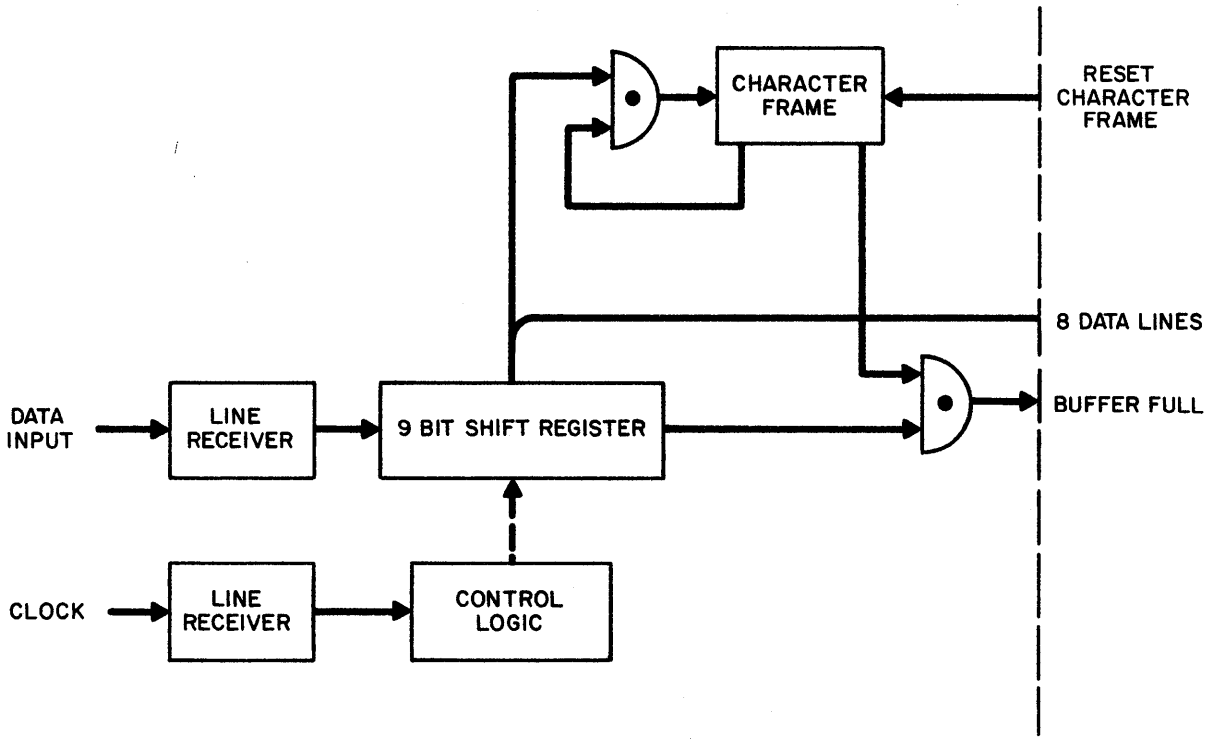


Figure A-1. Synchronous Input Line Interface Module

program desire to again cause a reframe state in the interface, it resets the Character Frame Flip-Flop.

A.2 SYNCHRONOUS OUTPUT LINE INTERFACE MODULE

The output interface module, which serves as a character buffer between the L-1192 and one output data channel, is presented in Figure A-2. The transmission of data from the line interface is synchronized with the transmit clock signal received from the data terminal bay. Each interface may operate at a transmission rate of up to 4800 band.

The character to be transmitted on the data channel is loaded into a 10-bit output shift register, in bit positions two through nine. The tenth bit is to set to "one", and the first bit serves as a buffer for unloading of the shift register. As the data character is shifted out of the register, "zeros" are inserted. When the bit inserted in the tenth bit position reaches the second bit position of the register, bits three through ten are "zero". A decode of this condition provides a "Buffer Empty" signal that the output interface is ready to receive another character. The L-1192 then loads another character into the output shift register and the above process is repeated.

A.3 ASYNCHRONOUS INPUT LINE INTERFACE MODULE

The asynchronous line interface modules will operate at up to 150 bits per second and require only a master timing signal, of N times the bit rate to set the speed of the device. The character input/output can vary from 5 to 8 level, since the interface shift register can accept 8 level plus start and stop.

The asynchronous output interface module is shown in Figure A-3. When the start bit is received, the clock is enabled and one-half bit time later, the data is shifted. At the same time the start-stop flip-flop is set, keeping the clock enabled.

If noise on the line causes the clock to start and the noise is present for less than one-half bit time, the clock is turned off and the counter is reset. Therefore, no shifting occurs.

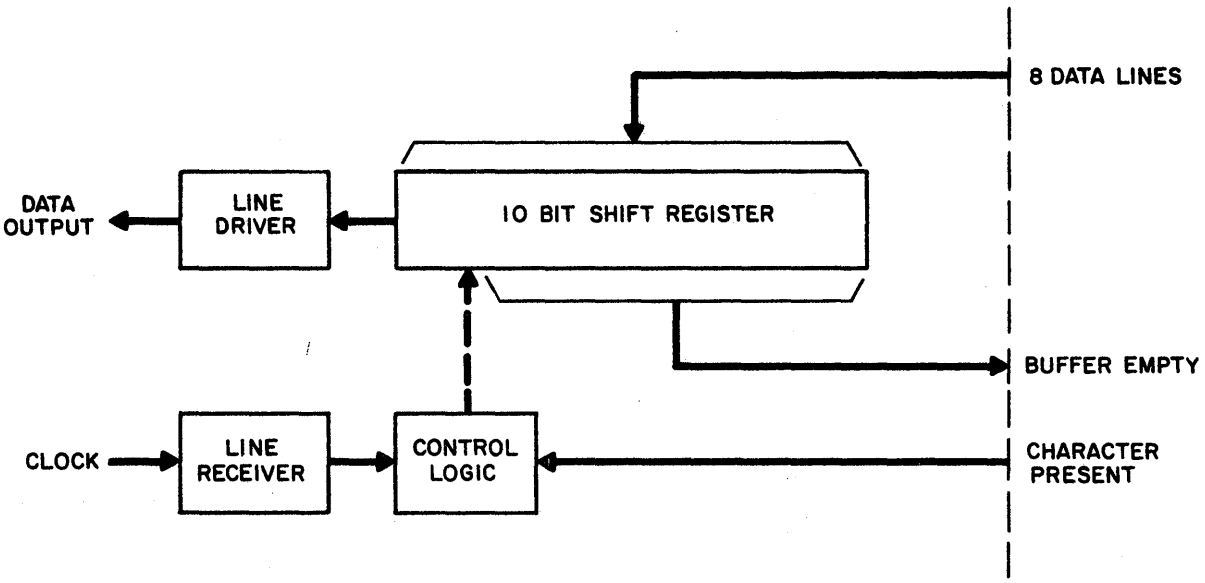


Figure A-2. Synchronous Output Line Interface Module

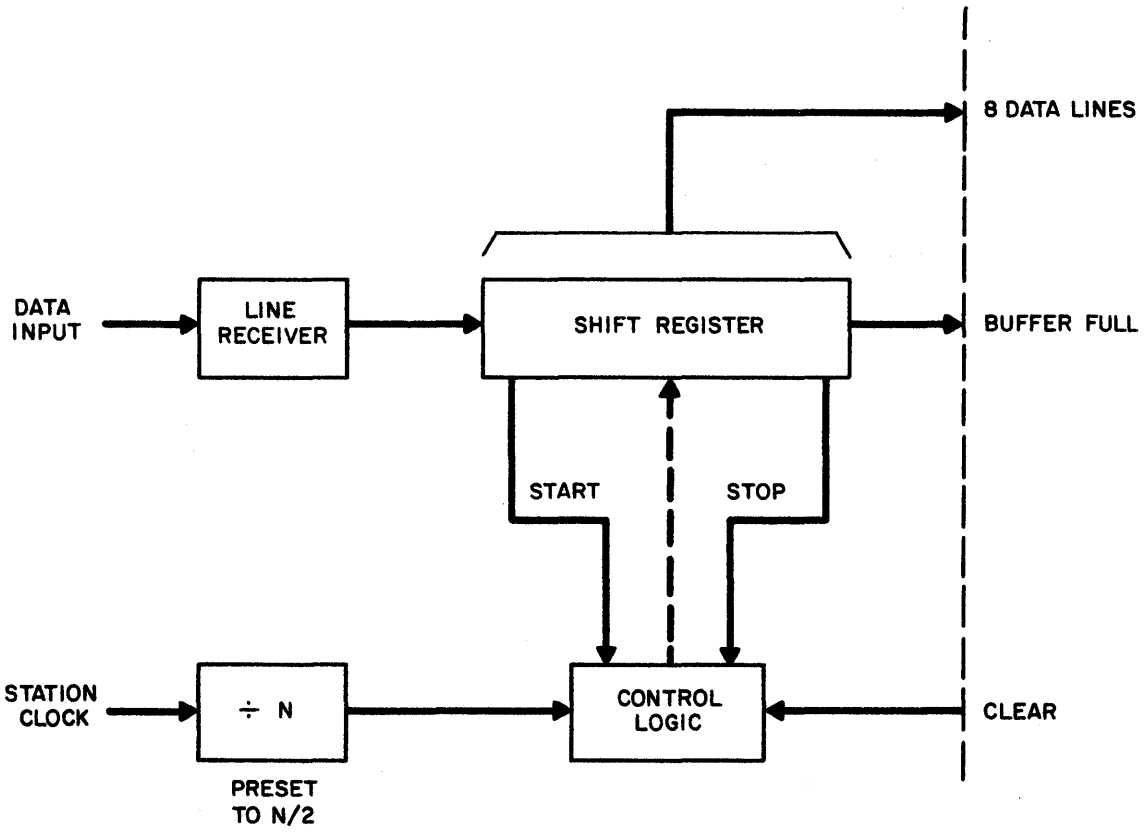


Figure A-3. Asynchronous Input Line Interface Module

When the start bit is shifted to the last bit position or indicator bit, it resets the start-stop flip-flop. As the stop bit is in the first bit position and the start-stop flip-flop is reset, the clock is turned off. At the same time a Buffer Full signal is sent to the L-1192. After the L-1192 reads the data lines, it sends back a clear signal which resets the shift register, preparing it for the next character.

A.4 ASYNCHRONOUS OUTPUT LINE INTERFACE MODULE

The asynchronous output interface module is shown in Figure A-4. When the interface is idle, a Buffer Empty signal will be present, and the L-1192 can generate a Character Present signal. The timing input is gated to the ÷ N counter. One bit time later, the pulse enable flip-flop is set. After another bit time, the data is loaded into the output shift register, causing the device enable signal to drop. The data is now transmitted, preceded by the start bit and followed by the stop bit. When the Stop bit reaches the output bit position, the device enable gate is energized and resets the ÷ N counter, the Pulse Enable Flip-Flop and the Character Present Flip-Flop.

A.5 DIALING FACILITY

The L-1192 Controller-Processor has been designed to make it adaptable to dial up out and dial answering facilities. The required Dial Converter and program procedures are dependent on the particular specifications of the designated automatic circuit switching system. The Dial Converter is the unit that converts digital subscriber selection information into the proper signals to perform circuit switching or addressing. The circuit switch is controlled by data transferred through the standard line interface modules.

A recent Librascope design, for example, interfaced a Controller-Processor with an AUTOVON dial switching system for synchronous transmission - as specified by the Defense Communication Agency. The Dial Converter for this specific application accepts digital subscriber addresses and converts them to dual-tone signals. The Controller-Processor

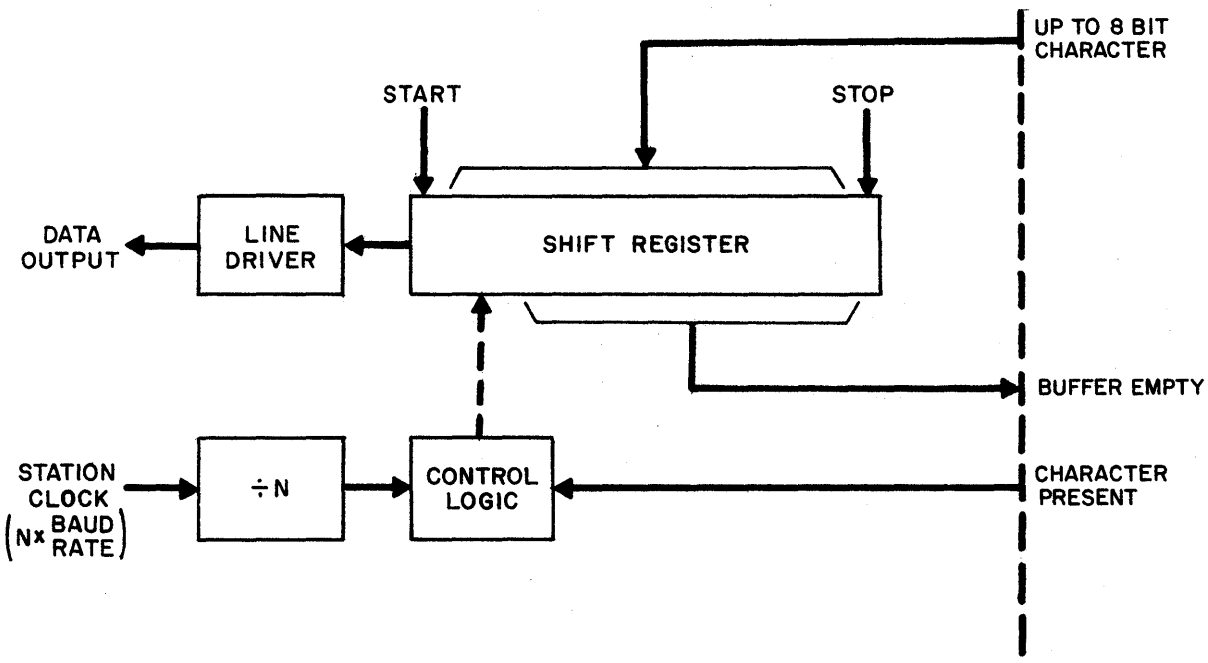


Figure A-4. Asynchronous Output Line Interface Module

transmits requests for service through a standard output line interface module. It also sets the characters frame flip-flop in the input interface, making it possible to detect the line going from continuous mark to a space condition--indicating acceptance of the request for service. The program then transmits a dial up call sequence, and waits for idle characters to signify establishment of link synchronization. If idle characters are not received, the call sequence is again initiated. After three attempts the station supervisor is notified. For dial answering, the line interface and program operates as in a direct station connection. The receipt of a character-frame character signals an input service request, and the interface automatically frames and transfers the subsequent characters.

Appendix B

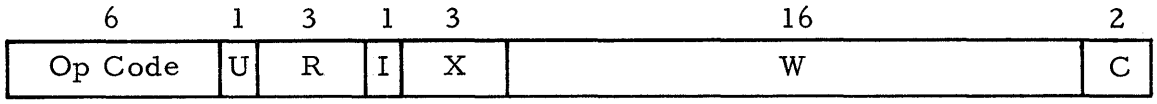
L-1192 INSTRUCTION SET

The following is a complete list and description of each L-1192 instruction. For ease in reference, it has been grouped according to instruction function.

The timing provided with each instruction includes instruction and operand access, and is given in memory cycles (nominally each 5 microseconds). If indirect addressing is employed, one cycle must be added for each indirect level. Indexing does not increase the instruction time.

The table shown below shows the general format for L-1192 instructions. Where an individual instruction deviates from this format, the change is described in detail.

L-1192 GENERAL INSTRUCTION FORMAT



U — Functions as described for individual instruction, or not used.

R — Operand 1 source. R = 0, 1, . . ., 7: use one of the eight data registers A, B, . . ., H directly.

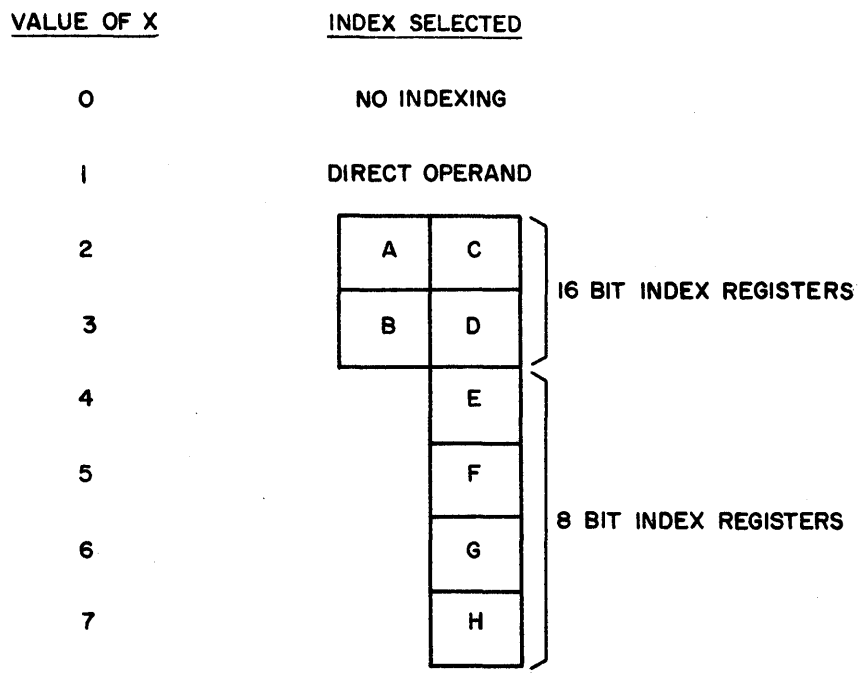
I — Operand 2 indirect bit. I = 0: direct. I = 1: indirect.

X — X=0: No indexing. (See Figure B-1).
X=1: Use the right-most bits of W/C as a direct operand.
X=2, 3: Use 16-bit index registers A, C and B, D.
X=4-7: Use 8-bit index registers E, F, G, H.

If W/C contains all zeros, X is used to specify operand 2 as noted below.

W/C — Operand 2 source. Obtain operand 2 from core memory word location W and character position C, indexed as specified by X,

CODING OF THE X FIELD



CODING OF THE R FIELD

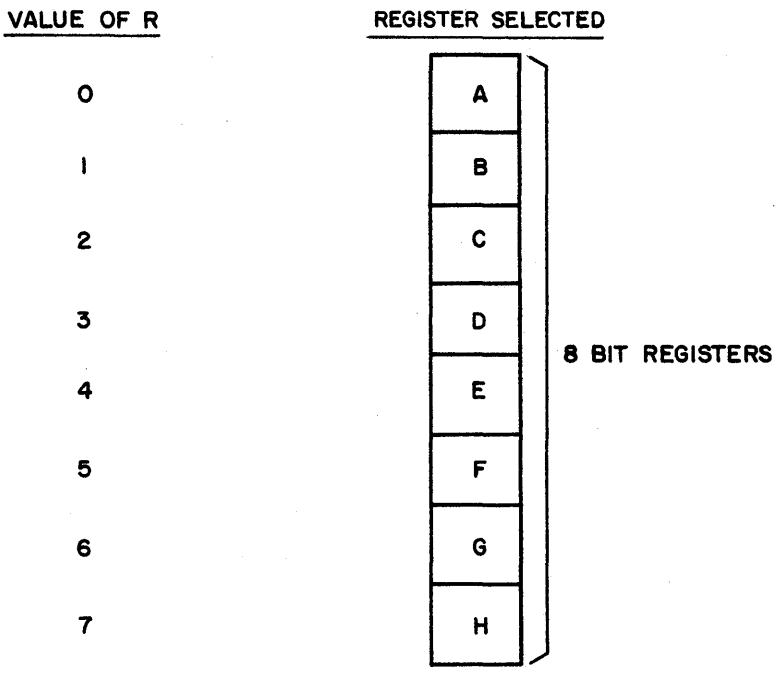


Figure B-1. Register Selection

and indirectly if specified by I. If W/C contains all zeros, operand 2 is obtained from one of the eight registers A, B, . . ., G corresponding to X = 0, 1, . . ., 7.

Note: For simplicity, operand 2 is usually referred to as "the specified character in memory."

B.1 FIXED LENGTH DATA OPERATIONS

These instructions include those for moving, comparing, modifying or performing arithmetic operations on fixed length data. These instructions are primarily used for performing character buffering functions. Most of them permit the selection of the data register in which the operation is to be performed. Those operations which address core memory allow indexing by any of the 6 index registers, and indirect addressing. If indirect addressing is specified, each indirect level may independently use indexing.

1. Character Movement

The following instructions permit single characters of data to be moved between the specified data register and the specified character position in core memory.

BRR Bring to R

Bring the specified character in memory to register R.
Timing: 2 IF X ≠ 1, 1 IF X = 1

BRM Bring to R through Mask

Bring the specified character in memory to register R through the mask in Register E. Zeros in E inhibit information transfer to the corresponding bits of R.
Timing: 2 IF X ≠ 1, 1 IF X = 1

STR Store R

Store the character in register R into the specified memory location.
Timing: 2

STM Store R through Mask

Store the character in register R into the specified memory location through the mask in register E. Zeros in E inhibit change in the corresponding bits of the memory location.
Timing: 2

2. Word and Partial Word Operations

The following instructions permit word and partial word operations dealing with the data registers. They are useful for rapidly loading or unloading a group of registers when changing processing states as when responding to an interrupt. The R field of these commands is an extension of the Op Code.

BRL Bring to Low Registers

Bring fields C1, C2, C3 and C4 of the specified word in memory to the low registers A, B, C and D respectively.
Timing: 2

BRH Bring to High Registers

Bring fields C1, C2, C3 and C4 of the specified word in memory to the high registers, E, F, G and H respectively
Timing: 2

STL Store Low Registers

Store the characters in the low registers A, B, C and D into the specified word in memory in fields C1, C2, C3 and C4 respectively.
Timing: 2

STH Store High Registers

Store the characters in the high registers E, F, G and H into the specified word in memory in fields C1, C2, C3 and C4 respectively.
Timing: 2

BXA Bring the right-most 16 bits of the specified word in memory to the data register pair A, C designated as index register 2.
Timing: 2 IF X ≠ 1, 1 IF X = 1

BXB Bring the right-most 16 bits of the specified word in memory to the data register pair B, D designated as index register 3.
Timing: 2 IF X ≠ 1, 1 IF X = 1

SXA Store the data register pair A, C designated as index register 2 in the right-most portion of the specified word in memory.
Timing: 2

SXB Store the data register pair B, D designated as index register 3 in the right-most portion of the specified word in memory.
Timing: 2

3. Character Length Arithmetic and Compare Operations

These instructions will perform unsigned arithmetic operations on a single 8-bit field, or will compare the contents of one of the data registers with a specified character in Core memory. All characters are considered to be magnitudes, and subtraction underflow will yield a complement remainder.

ADD Add

Add the specified character in memory to register R and put the result in register R. Overflow sets the overflow memo.
Timing: 2 IF X ≠ 1, 1 IF X = 1

SUB Subract

Subtract the character in memory from register R and put the result in register R. Underflow sets the overflow memo.
Timing: 2 IF X ≠ 1, 1 IF X = 1

CCM Compare Character to Memory

The character in register R is compared to the specified character in memory. If the register contents are greater. than memory, the next sequential instruction is executed. If the two characters are equal, one instruction is skipped. If the character in memory is less than the register contents, two instructions will be skipped.
Timing: 2 IF X ≠ 1, 1 IF X = 1

4. Logical Operations

The L-1192 logical operations enable setting and testing any combination of bits held in any of the data registers.

EOR Exclusive Or

Replace the specified character in memory with its bit-by-bit exclusive-or with the contents of register R.
Timing: 2

SCH Set Character

Set all selected bits of register R to one (U = 1) or zero (U = 0). Selected bits are those which are masked by corresponding ones in W. (I is not used.)
Timing: 1

TSC Test Character

Test all selected bits of register R for values of one (U = 1) or zero (U = 0). Selected bits are those which are masked by corresponding ones in M. A skip occurs if (1) I = 1 and all selected bits match U, or if (2) I = 0 and a mismatch exists.
Timing: 1

TSP Test Parity

Test all selected bits of register R for parity. Selected bits are those which are masked by corresponding ones

in M. A skip occurs if (1) I = 1 and parity is odd, or (2) I = 0 and parity is even. (U is not used.)
Timing: 1

5. Shift Operations

Single characters may be shifted within any of the data registers. Shifting may be open or cyclic as specified. The number of places shifted may be modified by an index register. The shift count (N) is contained in the right-most bits of the instruction. A maximum of 8 shifts will be made.

SCL Shift Character Left

The contents of register R are shifted left the number of places specified by N. Bits shifted past the high order end of the register are lost, and zeros replace those shifted away from the low order positions.
Timing: (N ÷ 4) + 1

RCL Rotate Character Left

The contents of register R are shifted left the number of places specified by N in a cyclic manner. Bits leaving the high order end of the register are shifted into the low order end.
Timing: (N ÷ 4) + 1

B.2 PROGRAM CONTROL INSTRUCTIONS

NOP No-Operation

Proceed with next instruction. (Op code field used only.)
Timing: 1

JMP Jump

Take the next instruction from memory location W as modified by X and I. (R and U are not used.)
Timing: 1

JSL Jump and Store Location

The location of this instruction plus one are stored in the memory location specified by W. The next instruction is taken from location W+1. Only the word address in location W is affected. Remaining positions are left unchanged.
Timing: 2

HLT Halt

Execution of this instruction halts the computer program by removing the RUN condition. Depressing the PROGRAM START pushbutton on the operators console restarts the program with the instruction specified by the program counter. (Op code field used only.)
Timing: 1

B.3 INDEX MODIFICATION

While the contents of any of the data registers may be modified by means of arithmetic instructions and tested by means of the character compare instruction, additional operations have been provided to allow the rapid and simultaneous modification and testing of registers. These instructions will allow the saving of significant amounts of time when executing short program loops.

The index modification instructions conform to the general format, with the exception that the fields have slightly different purposes. The X field specifies the register to be incremented or decremented. The R field specifies either the amount by which to modify the register (U = 0), or the data register which contains the modification value (U = 1). If I = 1, the contents of the register will be tested as the modification occurs with a program branch as the possible result. If I = 0, no test or branch will be made.

DXR Decrement Index Register

The contents of the instruction R field (U = Ø), or the contents of the register specified by the R field is subtracted from the index register as specified by X. If I = Ø, the next sequential instruction will always be executed. If I = 1, the result of the decrement will be tested. If the decrement did not cause the index register contents to pass through zero, the next instruction is taken from the location specified by W. If the index register did pass through zero, the next sequential instruction is executed.
Timing: 1

IXR Increment Index Register

The contents of the R field or the register specified by R, according to the setting of U, is added to the index register specified by X. If I = Ø, the next sequential will be executed. If I = 1, the results of the increment will be compared to W/C of the instruction. If the index is greater, or if it overflowed as a result of the increment, the next sequential instruction will be executed. Otherwise one instruction will be skipped.
Timing: 1

B.4 VARIABLE FIELD OPERATIONS

In addition to processing individual characters, the L-1192 can also perform operations on variable length fields. In these variable field operations, each of the two operands may have independent addresses and field lengths, up to a maximum of 16 characters. The addresses of both operands are subject to normal modification via indexing and indirect addressing.

A separate register, called the Field Operand Address Register, is provided to hold the address and length of the first operand. The contents of this register may be set or stored by means of special instructions.

Variable field arithmetic operations may be performed on both binary and decimal operands. The sign of each operand is carried in the least significant character field. These arithmetic operations are algebraic, and are executed as described below.

Decimal:

The least significant character of each operand field contains the sign. The four least significant bits of each character field contain the decimal digit. Bit 6 contains the sign. The remaining bits will be ignored. Operand 1, whose address is in the Field Operand Address Register, is added to or subtracted from operand 2, whose address is carried in the actual field instruction. The result of the operation is stored in place of operand 2. The result of the operation will be in true sign and magnitude form. If the results passed through zero, a ten's complement cycle will be taken to restore the result to true form. To yield meaningful results in the event the operand lengths are not the same, operand 2 must have the greater length.

Binary:

The least significant bit of the least significant character of each operand is considered to be the operand sign. Processing is similar to that described for decimal operands above, and operand 2 must always have the greater length.

*Note: The timing for field instructions is normally 1 + F1 + F2 cycles, where F1 and F2 are the length of the first and the second fields respectively. If a recomplement cycle is required, add another F2 cycles.

IFO Initialize Field Operation

This instruction sets up the Field Operand Address Register (FOAR). R and U are considered to be a count, and are transferred directly to the FOAR. The W/C field remains in the instruction register until the specified address modification has been completed, at which time W/C is transferred to the FOAR. Except for the special case of

the Compare Fields Equal instruction described below, the FOAR must be initialized each time before another variable field operation is performed.

Timing: 1

SFA Store Field Address

The contents of the FOAR are stored in R, U, and W/C of the specified memory location. The R field of the instruction is a part of the operation code.

Timing: 1

AFB Add Field Binary

The two binary operands whose addresses are specified by the FOAR (operand 1) and W/C of the instruction (operand 2) are added algebraically. The results of the addition occupies the position of the second operand. The result will carry true sign and magnitude. Overflow will set the overflow memo.

*Timing: See Note

SFB Subtract Field Binary

The two binary operands whose addresses are specified by FOAR (operand 1) and W/C of the instruction (operand 2) are subtracted algebraically. The remainder occupies the location of the second operand. The remainder will carry true sign and magnitude. Underflow will cause the overflow memo to be set.

*Timing: See Note

AFD Add Field Decimal

The two decimal operands whose addresses are specified by FOAR and W/C of the instruction are added algebraically. The sum is stored in the location of the second operand. The sum will carry true sign and magnitude. Overflow will set the overflow memo.

*Timing: See Note

SFD Subtract Field Decimal

The two decimal operands whose addresses are specified by FOAR and W/C of the instruction. The remainder is stored in the location of the second operand, and carries true sign and magnitude. Underflow will set the overflow memo.

*Timing: See Note

CFE Compare Fields Equal

This instruction compares the two fields whose addresses are specified by FOAR and W/C of the instruction. The field lengths are assumed to be identical, and the length count in FOAR will be ignored. Comparison is on a bit-by-bit basis, with signs not considered. Equal comparison causes the next sequential instruction to be executed. Unequal comparison will cause one instruction to be skipped. Since FOAR need not be initialized before each execution of CFE, automatic indexing through a table of contiguous entries may be achieved without the use of index registers.

*Timing: See Note

CFM Compare Field Magnitudes

The operands whose addresses are specified by FOAR and W/C of the instruction are compared arithmetically with signs treated as part of the magnitude. If the operand addressed by FOAR is greater than or equal to the operand specified by W/C of the instruction, the next sequential instruction will be executed. If not, one instruction will be skipped. The two operands used with CFM need not have identical lengths.

*Timing: See Note

MFM Move Field in Memory

The field whose address is specified by FOAR is moved to the address specified by W/C of the instruction. Reaching

the end of the shorter field will terminate the operation.

*Timing: See Note

B.5 COMMUNICATION INTERFACE INSTRUCTIONS

Due to frequency of servicing the communication interface, instructions are provided which greatly reduce the processing load when removing data from the low and high priority channel activity tables (CAT 1, CAT 2). In a single instruction are combined the functions of testing for the presence of data in CAT 1 or CAT 2, bringing the next entry to be processed to the data registers if data is waiting to be processed, and moving hardware bookkeeping markers to the next entry to be processed. (Refer to section 5.3.2.4 for the format of CAT 1 and CAT 2.)

When either ICH or ICL is executed, the relative positions of the load and unload markers are tested. If the same, indicating the tested table is empty, the instruction terminates and the next sequential instruction is executed. If the markers are not the same, the entry at the location of the unload marker (U1, U2) is brought into the data registers as shown:

Character	E Register
Channel Identity	F Register
I/O Indicator	G Register

The unload marker is then stepped cyclicly to the next entry position in the table. The next sequential instruction is skipped and this instruction is terminated.

ICH Initiate High Priority Character Cycle

The relative positions of LI and U1 are tested. If identical the instruction terminates. If not identical, the next entry in CAT 1 is obtained from the location specified by U1. The next sequential instruction is skipped and U1 is automatically stepped.

Timing: 1 IF L1 = U1, 2 IF L1 ≠ U1

ICL Initiate Low Priority Character Cycle

The relative positions of L2 and U2 are tested. If equal, the current instruction terminates and the next sequential instruction is executed. If unequal, the next entry is accessed from CAT 2 at the location specified by U2. U2 is stopped, the following instruction is skipped, and the current instruction is terminated.
Timing: 1 IF L2 = U2, 2 IF L2 ≠ U2

B.6 INPUT/OUTPUT INSTRUCTIONS AND COMMANDS

The detailed operation of the I/O Interface is described in section 5.5.3. For the purpose of the following discussion, instructions are those operations which are decoded and executed in the L-1192 instruction register. Commands are those operations which are executed in the I/O interface.

To initiate I/O operation, it is necessary for the L-1192 program to tell the I/O interface where the first of a list of interface commands will be found. This instruction will load the Interface Command Address Register (ICAR) with the location of the first of a chain of commands. The interface will then proceed to execute this list of commands independently of the main L-1192 program. At any time, various parameter which are contained in the interface registers may be accessed by the main processor program and examined. Direct output to some of the interface registers is also possible, thus providing the main processor program with the capability of forcibly terminating an I/O operation in progress.

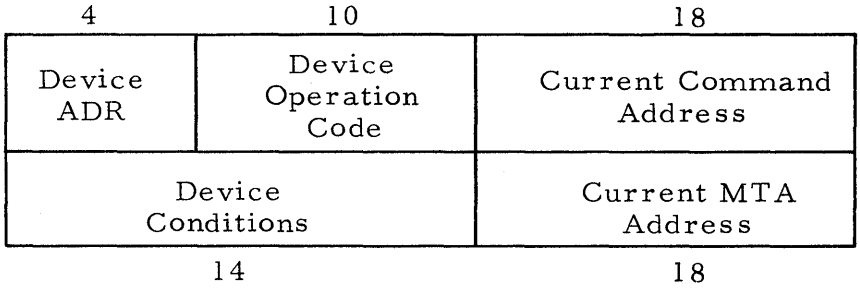
IIO Initiate I/O Operation

The contents of R and U selects the interface to be placed in operation. The W field of the instruction, after any specified address modification has been completed, is transferred to the ICAR. If the interface selected is already in operation, the instruction executed is NOP and the next sequential instruction is executed. If the interface is not busy, the instruction executes as described above, and the following instruction is skipped.
Timing: 1

SIO Save I/O Registers

Two words, as described below, are transferred from the interface specified by R, U to core memory at the location specified by W/C.

Timing: 3



Data Format for SIO

HIO Halt I/O Operation

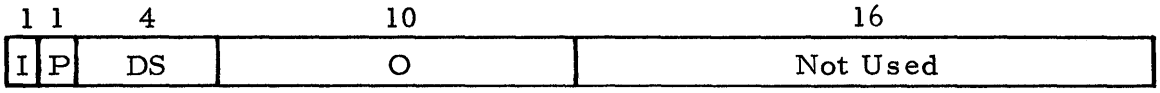
The execution of this instruction will cause the interface specified by R, U to cease operation immediately. The contents of the interface Block Control Register and the interface Command Register will be destroyed by this instruction.

Timing: 1

Interface Commands

The command described below are executed by the interface. The interface command must be the first word accessed when the interface is placed in operation. Only the general form of the command is discussed. The meaning of each Device Operation Field value and the relationship between device number and actual device will be determined by device operating characteristics and customer requirements.

Interface Command Format



I - Command indicator. Always one.

P - Proceed indicator. For operations which do not require the movement of any data the P bit will be zero, causing the interface to disconnect when the command is executed. If P is a one, the word following the command will be accessed.

DS - Device Selection. The value appearing in this field determines which I/O device assigned to the interface will be selected.

O - Device Operation Code - The exact assignment of binary codes to device functions is dependent upon the device itself. The translation between the device operation code and the actual device function controls will be largely done by the device adapter.

B.7 INTERNAL COMPUTER CONDITION PROCESSING

While the L-1192 is a relatively simple processor, it nevertheless has a number of internal condition indicators which must be tested and manipulated. Included among these conditions are the various machine interrupts, and devices such as memo indicators, breakpoint switches, and status and error indicators.

Interrupts

The interrupts are enabled through the use of a mask, which may be stored anywhere in core memory. The bits within this mask correspond to the various interrupt conditions. By enabling interrupts from this mask, only the desired interrupts may be used.

ESI Enable System Interrupts

The word at location W is used as a mask to enable interrupt operation for the I/O Interface specified by R, U. For every position which contains a one, the corresponding interrupt will be enabled. For every position which contains a zero, the corresponding interrupt will be disabled. The assignment of bits to interrupts is shown below.

Timing: 2

INTERRUPT ENABLE BIT ASSIGNMENT

BIT	FUNCTION
32	Interval Timer Reached Zero
31	I/O Operation Complete
30	Own Disc Requests Attention
29	Disc Assigned to 7090 Requests Attention
28	1460 Computer Requests Service
27	Operator Interrupt

System Conditions

System conditions may be set or reset, and tested for on or off condition by means of a pair of instructions. These instructions utilize the W/C field to select the memo or condition to be set or tested. This memo or condition is therefore subject to index register modification.

SSC Set System Condition

For those conditions which can be switched by program, this instruction will cause the state of the addressed device to be equal to the status of the U bit. Thus, if the U bit is zero, the device will be reset. If U is one, the device will be set. If this instruction is addressed to a device which is not program settable, it will execute as NOP.

Timing: 1

TSC Test System Condition

The condition of the specified memo or switch is compared to the U bit. If the condition and the bit are equal, the next sequential instruction will be executed. If not equal, one instruction will be skipped. The following table shows the addresses for the standard addressable conditions and devices. Additional devices may be added at customer option.

Timing: 1

DEVICE ADDRESSES

Address (in Octal)	Device
001	Overflow Memo
010-017	Breakpoint Switches (4 standard)
020-027	Memo Lights (4 standard)
100	Master Inhibit Interrupt, I/O Interface 1
101	I/O Interface Busy, I/O Interface 1
300	Lock Communication Scanner on L-1192 Paper Tape
301	Lock Communication Scanner on Low Speed I/O Interface
310-347	Read or Write Low Speed I/O Devices

Appendix C

PROGRAMMING DESCRIPTIONS

C. 1 LAP-1192 ASSEMBLER

C. 1. 1 Purpose

To accept a source program consisting of symbolic instructions, and to generate an object program in machine code plus a side by side listing showing the original source instructions and the resulting machine code.

C. 1. 2 Equipment Required

a. L-1192 processor with at least one module (8, 192 words) of core memory.

b. At least 640, 000 additional characters of disk storage.

c. Typewriter, paper tape reader, paper tape punch.

C. 1. 3 Input Media

Symbolic instructions are punched into paper tape in card image (80 character records) by any off line method such as a Flexowriter.

C. 1. 4 Output Media

The object program listing is generated on the console typewriter. The object program outputs in a special code to the paper tape punch on the Operator Console. This code contains relocation and parity information, in addition to machine instructions.

C. 1. 5 Features

1. Allows compilation of large programs through the use of disk memory for storage of source and objects codes and the symbol table.

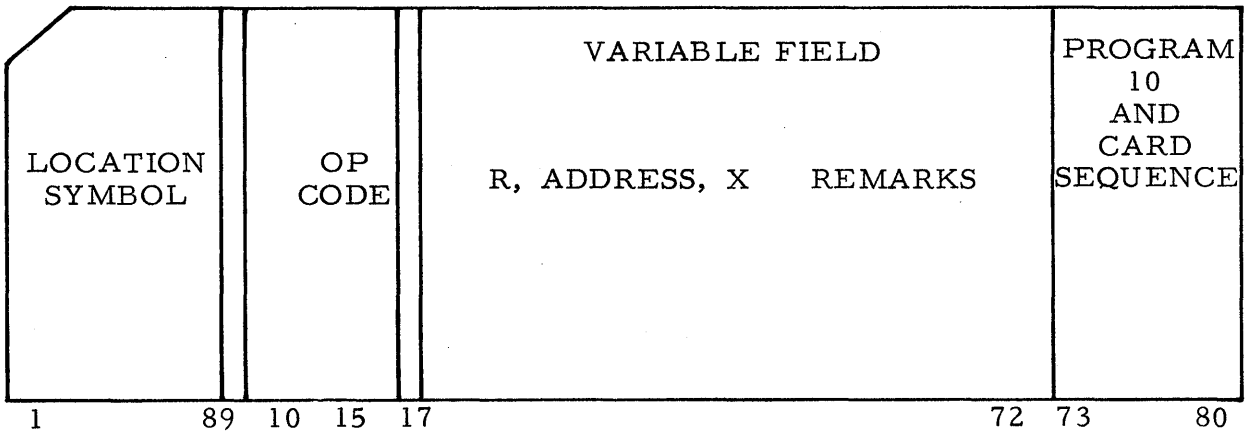
2. Permits absolute and relocatable addressing to be used at the option of the programmer.

3. Since symbol tables may be output and later read back as an option, large programs or systems of programs may be compiled in segments.

C.1.6 Input Format

The input is described here in card format, since card to paper tape is a common method for generating input tapes.

CARD FORMAT



1. Location Symbol

   This field may contain a symbolic address of up to 8 characters in length. Symbols may consist of any combination of letters and numbers, with the restriction that the left-most character must be a letter. A symbol in this field of an instruction card will enter the symbol table with the current value of the location counter.

2. OP Code

   The operation code must be punched left justified in this field. If indirect addressing is desired, a (*) is punched immediately after the last character of the operation code.

3. Variable Field

This field contains the parameters to be placed in the R, W/C, and X fields of the instruction. In the event the operation code defines the R field, this parameter must not be coded. The first parameter defined in this case must be the address.

All fields may contain either symbolic or absolute data. Absolute data will be treated as decimal. The parameter (*) by itself will result in the use of the current contents of the location counter for the value. Address arithmetic limited to addition (+), subtraction (-) and multiplication (*) is allowed.

After the last parameter of the variable field, a blank must appear. Following this, any desired remarks may be punched. These will appear on the output listing.

4. Program ID and Card Sequence

Any desired information may be punched into these columns.

C.1.7 Pseudo-Operations

In addition to assembling machine instructions, the LAP-1192 Assembler will also accept and process a list of pseudo-operations. These are used to control the assembler and to perform special operations with the object program. Each of these pseudo-operations is described below.

1. ORG

Sets the location counter to the value specified in the variable field. If the location field contains a symbol, it will be defined as the location counter values after setting. The parameter in the variable field may be octal, decimal, or symbolic. The normal parameter is octal. A decimal value is preceeded by (D/). A symbolic value consists of a previously defined symbol, and does not contain any special characters.

2. END

Ends the acceptance of the source program and starts the second pass of the assembler. If the variable field contains an address, the object program will contain information to cause a branch to that address when loading is completed. If no address is supplied, loading will halt when the program has been loaded.

3. DEF

The symbol contained in the location field is defined in terms of the parameter in the variable field. If the variable field contains a number, the defined symbol will always be absolute. If the variable field contains a symbol, the symbol in the location field will be defined according to the following process:

a. If the variable field is preceded by (A/), the defined symbol will unconditionally be absolute.

b. If the variable field is preceded by (R/), the defined symbol will unconditionally be relocatable.

c. If the variable field is not preceded by any specifier, the defined symbol will be absolute or relocatable according to the evaluation of the variable field.

4. OCT

The next location in the object program will contain the octal quantity appearing in the variable field.

5. DEC

The next location in the object program will contain the octal equivalent of the decimal expression appearing in the variable field.

6. BCD

The first 4 characters punched in the variable field will be stored in the next location in the object program in IBM BCD code. Illegal characters will be translated as blanks.

7. HEX

The 8 hexadecimal characters punched in the variable field will be placed in the next location in the object program. In hexadecimal code, the first 6 letters of the alphabet are used for those values above 9.

8. VFD

The variable field is evaluated and used to make up an L-1192 word. Each subfield is set off by a comma. The general format of the variable field of a VFD operation if VFD Fn/P_1, Fn/P_2 ..... Fn/P_i, where:

a. F: The format of the subfield
   D = Decimal
   O = Octal
   B = BCD
   H = Hexadecimal
   S = Symbolic

b. n: The number of bits in the field

c. P_i: The parameter to be placed in this subfield.

The subfield identified as P_1 will occupy the most significant bits of the next location of the object program. If more bits are specified than exist in a single word, the VFD operation will extend on to additional words as required.

9. BLK

A space of the size specified by the expression in the variable field will be reserved at this point in the object program. If a symbol is used in the variable field, it must be previously defined.

10. PST

The symbol table will be punched out at the end of the first pass. It will be in suitable form for input to LAP-1192 as described in the following operation.

11. RST

The read symbol table operation is not currently available to LAP-1192. It may be simulated by turning on breakpoint switch No. 1. If this is done, the symbol table must be input before the symbolic souce program is read in.

12. ABS

Produces an absolute address object program.

13. REL

Produces a relocatable address object program.

14. REM

The entire variable field will appear in the output listing as a remark. It will not affect the object program.

15. TITLE

If the first one or two cards of the source deck carry the OP code of TITLE, then variable fields will appear at the top of each page of the output listing as a program title. If the TITLE pseudo-op appears anywhere else in the source deck, it will be treated as REM.

C.2 I/O SERVICE ROUTINE

The I/O Service Routine is written to control all data transfers between core memory and the devices assigned to the I/O Interface. Programs desiring data transfers communicate their requests to the I/O Service routine by means of cyclic request tables. The data transferred is also intended to be stored in cyclic tables, allowing a degree of asynchronism between the execution of data transfer requests and the operation of the requesting programs.

C.2.1 Cyclic Tables

While the concept of the use of cyclic tables is quite common in data processing, this section will describe in detail how tables must be constructed for use by the I/O Service Routine.

A cyclic table contains a series of entries, which may be either single or multi-word. The table is treated as though circular, with the first entry immediately following the last (highest addressed) entry. The table must be an integral number of entries in length, so that the location of entries will not precess as the table is filled and emptied.

A list of parameters and bookkeeping markers associated with the table must be maintained. This parameter and marker list may be stored anywhere in memory. Within the list for a particular table, however, the location of each word must conform to the format shown as follows:

Word	Identifier	Contents
1	F	The address of the first word of the table.
2	L	The last address of +1 of the table.
3	I	The address where the next entry will be stored.
4	O	The address of the next entry to be removed.
5	C	The count of active words in the table.

Cyclic tables are processed according to the following procedures:

a. Whenever an entry is placed in the table, it is stored starting at the location of the I-marker. The I-marker is moved up by the number of words in an entry, and the C-counter is increased by the size of the entry.

b. Whenever an entry is removed, it is taken from the location of the O-marker. The O-marker is moved up by the number of words in an entry, and the C-marker is decreased by the size of the entry.

c. Whenever I or O is counted past L, it is recycled to F.

d. Whenever C = Ø, the table is empty. The value of C must always be tested before storing data in a cyclic table. If C is equal to

the difference between F and L, the table is full. Placing another entry in the table at this time will result in overwriting the oldest unprocessed entry.

C. 2. 2 Communication with I/O Service Routine

The I/O Service routine uses a separate cyclic request table for each I/O device attached to the I/O interface. This permits executing requests for transfer on a priority-by-device basis, without violating cyclic table processing rules. Each request for transfer consists of a 3-word entry, which has the following format:

Word	Contents
1	The location of the I/O interface command for this transfer.
2	The location of a count of the total number of words to be transferred.
3	The location of the first word (F) of the parameter list for the table involved in the data transfer. Word 3 must also contain an I-bit of 1, and an X field = 7.

C. 2. 3 Transfer Request Processing

Requests for transfer are executed on a priority basis. Transfers to or from the L-1192 disk memory receives highest priority, and transfers to or from the 7090 disk and the 1460 receive lower priority in descending order respectively. The I/O Service Routine will first examine the request table for the 1301 Disk File. If empty, the next lower priority device table will be tested, until an unexecuted request if found or it is determined that there are no more reports. The request will be executed if the I/O Interface is not busy, and the I/O Service Routine will then store the location of the request and exit.

Whenever the I/O interface goes "not busy" it interrupts the main cycle. The I/O routine is then entered, and interface conditions are stored for later reference. If the transfer aborted due to errors, it will attempt to re-execute the request up to a maximum of 3 times. If the transfer

cannot be executed successfully, the device will be set unavailable, and notice given to the operator. If the transfer executed without errors, the last two words of the applicable entry in the device request table will be used to update the bookkeeping markers for the table involved in the transfer. The I/O Routine will not check for overwrite of this table. It is the responsibility of the requesting program to insure that it has not requested an excessive amount of data.

When bookkeeping for the request just processed is completed, the markers for the device request table, which contains the completed request, will be updated. The I/O Routine will then begin another scan for unprocessed requests.