GENERAL PRECISION | LIBRASCOPE GROUP

808 WESTERN AVENUE, GLENDALE 1, CALIFORNIA

Technical Description

L-1192 CONTROLLER PROCESSOR

Technical Description

L-1192 CONTROLLER PROCESSOR

SURFACE EQUIPMENT DIVISION

GENERAL PRECISION INC.

LIBRASCOPE GROUP

808 WESTERN AVENUE • GLENDALE, CALIFORNIA 91201

CONTENTS

CONTENTS - Continued

CONTENTS - Continued

ILLUSTRATIONS

Section 1

INTRODUCTION

This document presents a technical description of the L-1192 Controller-Processor system. The Librascope design approach is based primarily on experience gained from their L-3055 data processing system. This equipment was developed and installed in the Pentagon as the AN/FYQ-11 data processing elements of Command and Control System 473-L. Extensive effort was expended to structure and create a system which will function reliably and continuously without loss of service.

The L-1192 system provides a stored-program binary computer, with both an AUTODIN communication interface and general purpose processing capability. It will operate as a communication buffer to the L-3155 Central Processor in the 473-L system, and can also provide this function for other available processing systems. The L-1192 Controller-Processor may also be utilized as a general-purpose computer on its own. It will interface with a variety of communication channels, input-output stations, and peripheral devices. These include asynchronous and synchronous lines, paper tape/card/typewirter I/O stations, magnetic tape transports, card reader/punches, disc files, and high speed line printers.

A description of the L-3155 Central Processor, which provides eight 7-bit alphanumeric characters per computer word, is provided under separate cover.

Section 2

SUMMARY OF APPROACH

The L-1192 Controller Processor is a general purpose computer system, with special emphasis on the efficient processing of input-output and communication data.

2.1 WORD FORMAT

The L-1192 provides 2 microsecond cycle cone memory in modules of 4096 words, with a maximum capacity of 65,536 words. The 33-bits per word includes 32 data bits, providing four 8-bit characters, plus one parity bit. This word length permits an instruction format well suited to character processing and addressing.

2.2 INSTRUCTION EFFICIENCY

The L-1192 offers three categories of processor instructions: single characted operations, variable-length field operations, and full-word operations. The single character instructions utilize a group of eight arithmetic registers, which eliminates the bottleneck normally encountered in processing multiple input-output data.

2.3 COMMUNICATION INTERFACE

The L-1192 Console can communicate with up to four duplex communication channels, by the addition of the proper interface modules. This capability can be expanded to 64 duplex channels by means of an optional Line Unit Console. Any mix of asynchronous and synchronous lines is permissible. Asynchronous lines operate up to 150 bits/sec, and synchronous lines at up to 4800 bits/sec.

2.4 CODE/FORMAT FLEXIBILITY

The L-1192 has been designed to efficiently process AUTODIN communication requirements. Care has been exercised to permit flexibility of

operation through program modification. Desired character codes, formats, and channel coordination procedures are under control of the stored program.

2.5 I/O STATIONS

The Standard Communication Interface will also communicate with up to eight input/output stations, which may be located at remote positions. One station is utilized by the 300 char/sec paper tape reader, 60 char/sec paper tape punch, and 15 char/sec electric typewriter at the Operator/Maintenance Panel. The other seven stations may each provide a 20 char/sec card read/punch and a 40 char/sec print station, or other desired I/O character device.
The number of I/O stations may be expanded in groups of 8 to a total of 32.

2.6 COMMUNICATION PROCESSING

Input characters and output requests for the communication channels and the I/O Stations automatically enter cyclic Channel Activity Tables reserved in core memory. These entries do not interrupt the system, and are accessible to the computer program for further processing.

2.7 I/O INTERFACE

One I/O Interface is provided with the L-1192 Controller Processor, and up to sixteen interfaces may be utilized. Each interface may communicate with up to sixteen standard Librascope peripheral devices. These include magnetic tape, card reader/punch, high-speed line printer, and disc file equipment.

2.8 SUPPORT SOFTWARE

The support software offered with the L-1192 system includes an Assembler, a System Loader, Core and Disc Dump, and Acceptance Test.

2.9 AUTODIN PROCESSING

Librascope has had extensive experience in processing communication data, and will be available to assist in meeting AUTODIN requirements. An operational approach developed for the L-1192 system is presented in this proposal.

Section 3

EQUIPMENT DESCRIPTION AND OPERATION

This section discusses the design and operation of each item of equipment available with the L-1192 Controller-Processor system.

3.1 CORE MEMORY

Core memory for the L-1192 Controller-Processor system is provided in modules of 4096 words. The L-1192 may address up to a maximum of sixteen 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 L-1192 instructions. Transfers to and from core memory are in word parallel, and these are parity checked. The core memory modules offer a read-write cycle time of 2.0 microseconds. The transfers between core memory and the central processor arithmetic/logical unit, the communication interface, and the input-output interface all overlap on a cycle stealing basis.

3.2 L-1192 CONTROLLER-PROCESSOR

The L-1192 Controller-Processor consists of an Arithmetic and Logical Unit, a Communication Interface, Paper Tape Reader/Punch and Electric Typewriter, an I/O Interface, and the Maintenance Panel, and an Interval Timer. The L-1192 Console is presented in Figures 3-1 and 3-2.

3.2.1 Arithmetic and Logical Unit

The L-1192 ALU includes the instruction register and instruction decode logic, individually addressable character-length data registers, a Field Operand Address Register, and two full-word arithmetic registers.

The simple yet flexible organization of the L-1192 is apparent from the block diagram of the processor organization, presented in Figure 3-3 any operations which involve two operands can be obtained in any combination from the data or word registers and memory by general bus logic. The

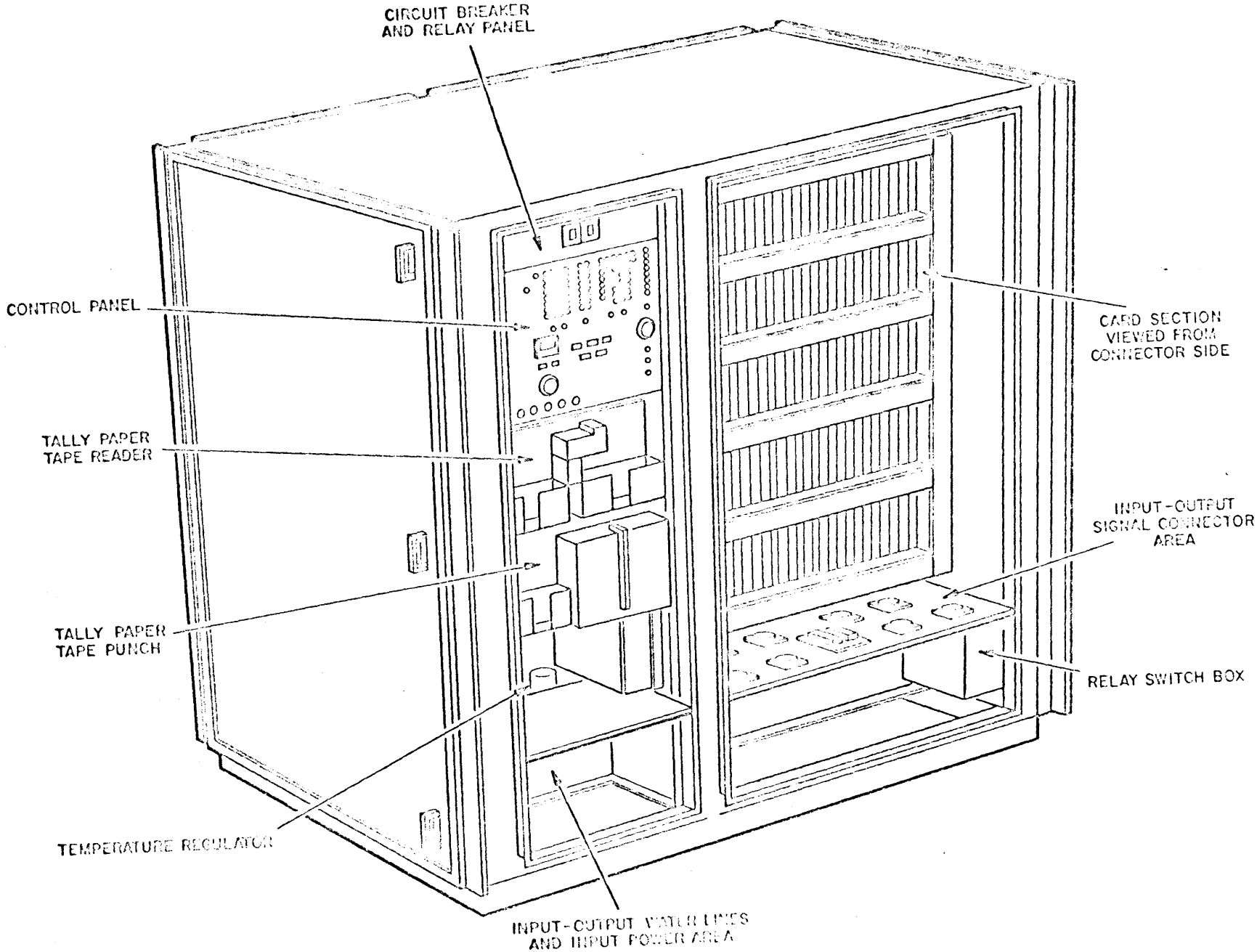


Figure 3-1. L-1192 Buffer Processor Console - Front View

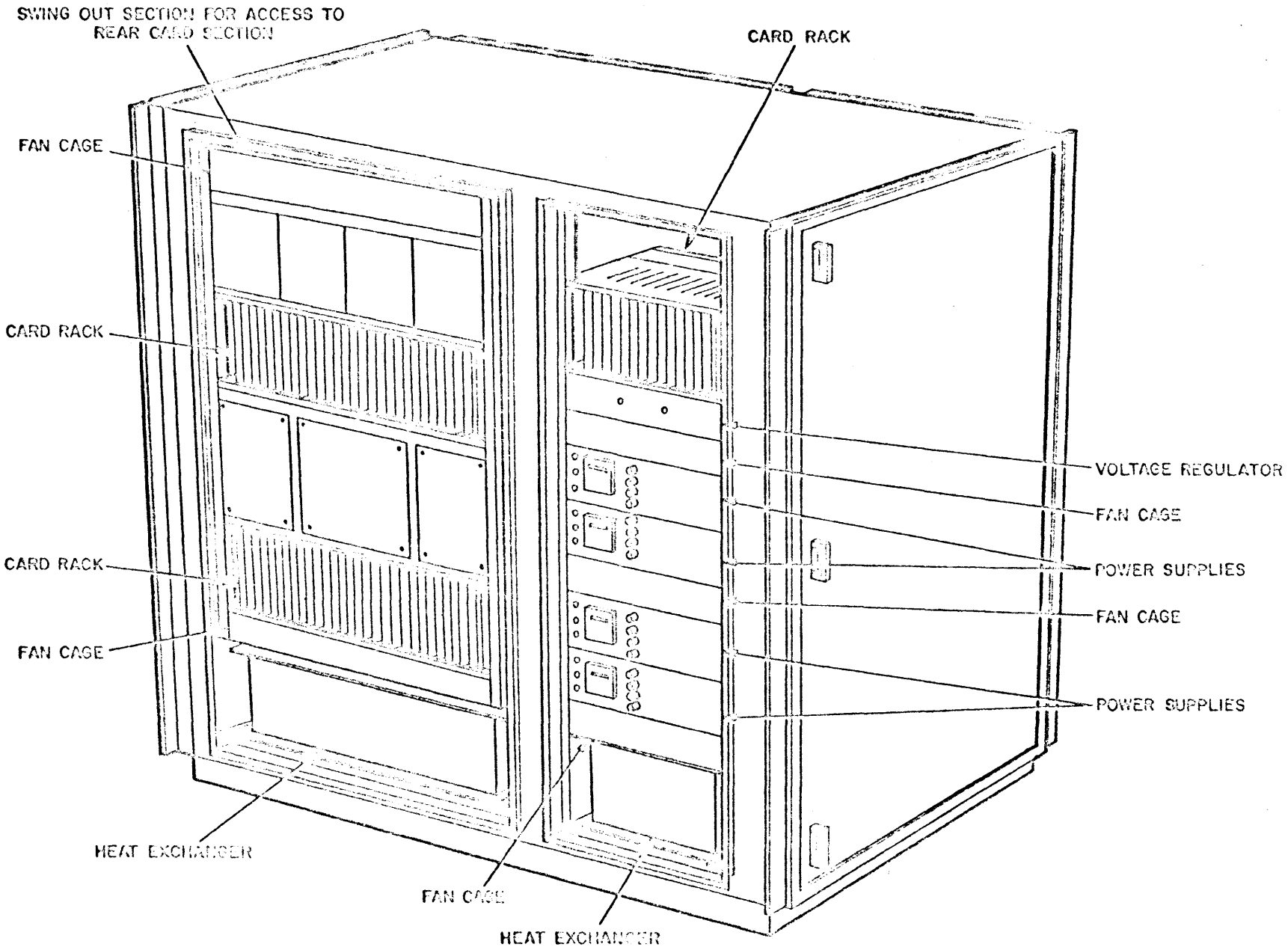


Figure 3-2. L-1192 Buffer Processor Console - Rear View

function of address modification is 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.

3.2.1.1 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, and these characters may be individually selected by the L-1192 instruction.

8	8	8	8	1
Character 4	Character 3	Character 2	Character 1	P

3.2.1.2 Instruction Format. Instructions are one word in length, and permit register selection in addition to memory word addressing. Special features of the instruction set include direct character selection, word and character indexing, indirect addressing for both memory and registers, and the inclusion of specialized instructions.

3.2.1.3 Memory Addressing. The address field of an instruction provides for direct reference to any one of 65,536 words of core memory. In addition, 2 bits appended to the low order end of the address field permit the selection of an individual character within the word. This feature provides full-character addressing.

3.2.1.4 Processor Instructions. The L-1192 offers three categories of processor instructions. Fixed length character instructions primarily provide single-character transfer, arithmetic, and logical operations. These facilitate the processing of data from multiple communication lines and direct-coupled paper tape, punched card, and typewriter/printer devices. A set of variable field (up to 16 character) instructions provide added formatting and editing capability for both communication and input-output devices. Word-length operations are also provided to meet higher speed processing requirements.

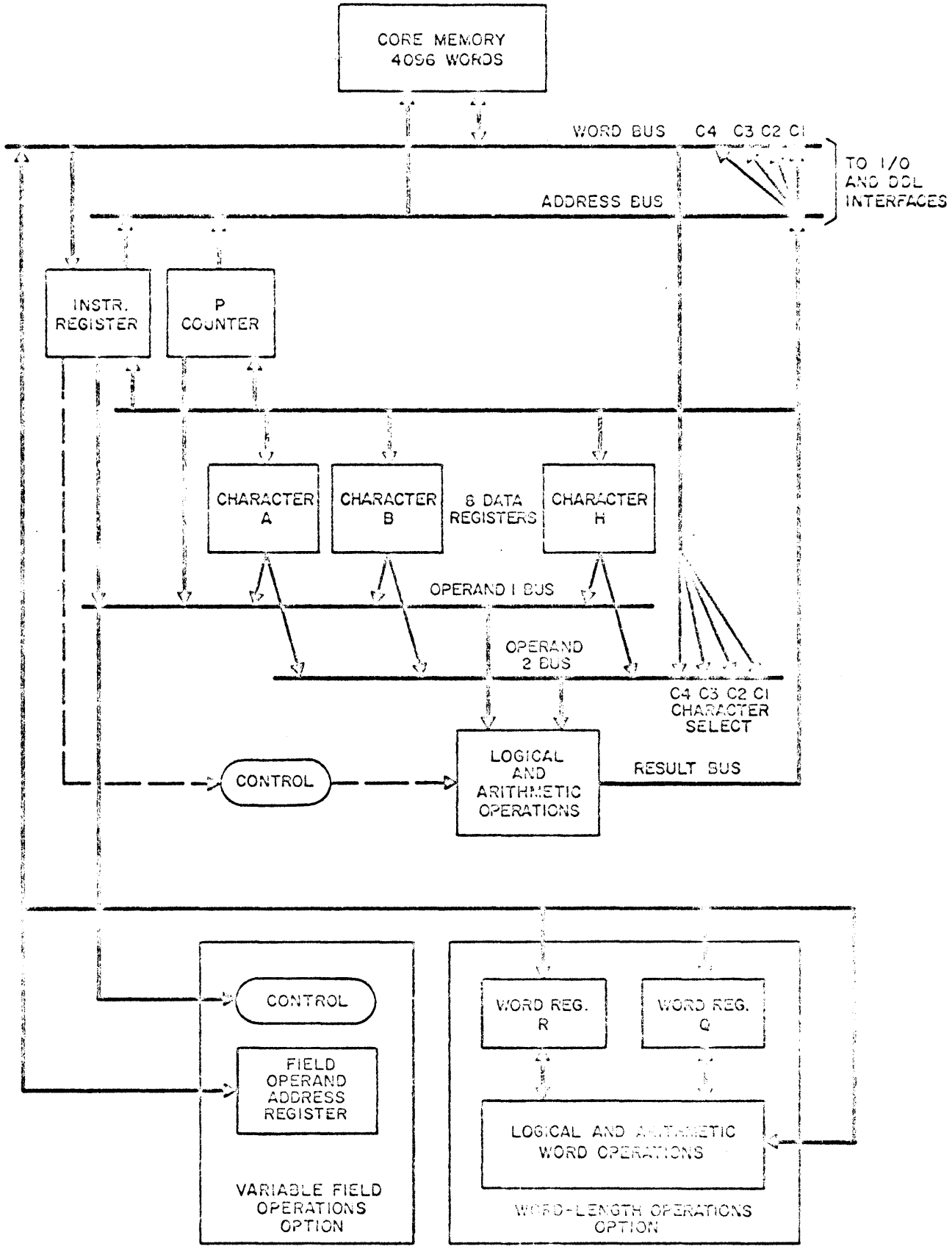


Figure 3-3. L-1192 Buffer Processor, Block Diagram

3.2.1.5 Specialized Instructions. Several of the L-1192 instructions have been specialized in order to enhance speed-cost effectiveness in communication and input-output processing. The inclusion of these instructions combine the advantages of wired logic with the flexibility of stored program control. An Initiate Character Cycle instruction, for example, will efficiently locate and access the next input character or output request to be processed. These were automatically stored in core memory Channel Activity Tables (CAT) by the L-1192 hardware.

3.2.1.6 Character Data Registers. The L-1192 is provided with a set of 8 character-length static registers, which are basic to its list of powerful character processing instructions and capabilities. Any of the 8 registers may be utilized in arithmetic, data, or logical operations. Most operations involve a single register and a location in core memory. Operations may also be specified which involve only a pair of registers, or a register and an operand or mask contained in the instruction word itself. These features serve to reduce the amount of time normally consumed by computers in the access and storing of operands in core memory. The unusually high character processing efficiency of the L-1192 is attained by making up to 8 separate data items immediately available to the arithmetic unit.

The data registers may also function as index registers. When used as index registers, four of the 8-bit 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 16-bit index registers, each with an addressing capacity of 65,536 characters. Indirect addressing is also a feature of the L-1192, and may extend over any number of levels. Indexing is effective at each indirect level.

3.2.1.7 Additional Registers. The variable field instructions provide memory to memory operations. One operand address and field length is normally specified by the instruction word. The second operand address and field length must first be loaded into a Field Operand Address Register (FOAR). The full-word instructions utilize two independent word

arithmetic registers (R and Q). The variable field instructions and the full-word instructions also utilize the character data registers as index registers. They are available as arithmetic registers only to the fixed length character instructions.

3.2.1.8 Error Checks. To insure message protection, the L-1192 employs several internal error checks. These include an automatic test of word parity when a word is transferred from core memory, and parity checks on transfer operations between the L-1192 and the I/O peripheral devices. Program testing is normally utilized to parity check characters received from the communication lines.

3.2.1.9 Interrupts. To permit efficient servicing of the Input-Output and Communication Interfaces, a number of interrupts have been provided. The conditions which can cause an interrupt include: Interval Timer elapsed, I/O operation ended, service requests from the peripheral devices, and internal or input-output parity error.

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 Ø, 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 PSI instructions.

3.2.2 Communication Interface

The Communication Interface is contained in the L-1192 Controller-Processor Console. It controls the transfers of data between core memory and on-line communication channels or devices.

The basic system can accommodate four input and four output communication channels, by the addition of optional Line Interface Modules. Modules may be specified for asynchronous or for synchronous transmission at rates up to 4800 baud. The communication interface capability can be expanded by the addition of an optional Line Unit Console, which can accommodate up to 64 input and 64 output communication interface modules and channels.

3.2.2.1 Core Buffering. 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.

3.2.2.2 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 Section 4.6.

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 Unit Console and in 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) or 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.

3.2.2.3 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.

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

3.2.3 Paper Tape, Typewriter, and Card/Print I/O

The basic Communication Interface also communicates with up to eight direct-connected input-output stations, which may be located at remote sites. The number of such stations accessible to the communication interface may be optionally increased in groups of eight up to a total capacity of 32 such stations. The paper tape reader, punch, and IBM keyboard/typewriter at the computer console provide one station position. A Kleinschmidt keyboard/printer, and a card reader/punch may be located at each of the other seven station positions. These stations will, however, accept other input-output devices which contain the proper

interface. Transfers to or from these devices are character-serial, with up to 8 bits/char.

3.2.4 I/O Interface

The I/O Interface provides for data transfers to and from the standard line of Librascope magnetic tape, card reader/punch, line printer, and disc file equipment. These peripheral devices are presently utilized in the Librascope AN/FYQ-11 Data Processor Set for the Headquarters U.S. Air Force 437-L Command and Control System. All data transfer 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 processor operations.

3.2.4.1 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 and a Block Control Word from core memory. The Command Word specifies the device and command operation. The Block Control word designates the memory starting address and the required number of characters. 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 the Block Control word is satisfied, the operation is terminated and the main processor program is interrupted.

3.2.5 L-1192 Maintenance Panel

The maintenance panel, presented in Figure 3-4, contains the necessary controls and indicators for performing preventive or corrective maintenance on the L-1192 Controller-Processor and its interfaces. By means of these controls and indicators, the various internal registers are displayed and their contents may be altered. In addition, L-1192 error

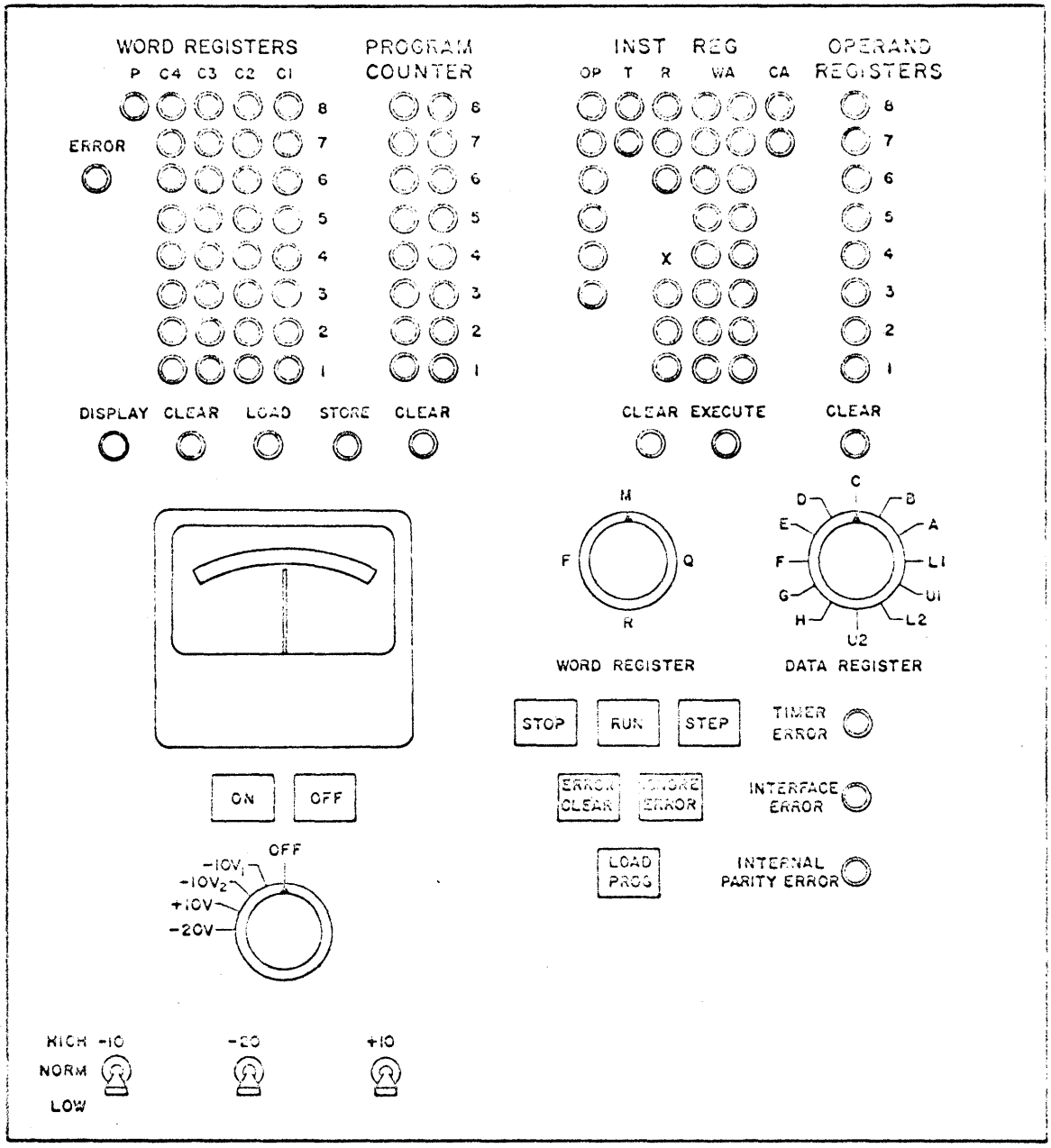


Figure 3-4. L-1192 Maintenance Panel

indicators may be reset from this panel. The output voltages of the various power supplies may be measured, and their outputs biased to assist in detecting machine malfunctions. The following paragraphs describe in detail the functions of the various controls and indicators on the maintenance panel.

3.2.5.1 Word Register Display and Switches. An array of 33 pushbutton indicator/switches displays the content of the specified word register. Selection of the Memory Access Register (M), the Field Operand Access Registers (F), or the word arthmetic R and Q Registers is by means of a "Word Register" select switch. The display is arranged into 4 characters of eight bits each and the parity bit. An associated error indicator is illuminated when the memory register is selected and contains a parity fault condition (even number of bits). Direct manual memory entry is performed using the Clear and Load pushbutton switches. The Clear pushbutton clears the selected register to binary zeros, except for the parity bit which is a binary one to maintain required odd parity. The desired word is then set by pushing appropriate register pushbuttons. Pushing the parity pushbutton will set the correct parity bit and reset the error indicator.

Pressing the STORE pushbutton stores the contents of the selected register into the address specified by the W portion of the Instruction Register. W is then incremented by one. The content of the memory location specified by W may be displayed by pressing the DISPLAY button, which also causes W to increment by one.

3.2.5.2 Program Counter Display and Controls. An array of pushbutton indicators displays the content of the program counter. The "PROGRAM COUNTER" display can be set to zero by the associated "CLEAR" pushbutton and new content set in manually.

3.2.5.3 Instruction Register Display and Controls. An array of pushbutton indicators displays by fields the content of the instruction register. The "INST REG" display can be cleared by the associated "CLEAR"

pushbutton and new content set in manually. The currently contained instruction can be executed in stop mode by pushing the "EXECUTE" button, permitting instruction processing without counting up the program counter or leaving the stop mode.

3.2.5.4 Operand Registers Display and Controls. The "OPERAND REGISTERS" display pushbutton indicators may be used to observe the content of any of the 8 Data Registers, or the content of the two Load and Unload Counters associated with the communication interface. Selection of register or counter to be displayed is controlled by a rotary "DATA REGISTER" switch below the "OPERAND REGISTERS" indicators. The register currently displayed may have its content cleared to zero by means of the "CLEAR" pushbutton and a new content manually entered.

3.2.5.5 Error Indicators. The three error indicators in the lower right-hand section of the panel display detail errors detected by the L-1192 hardware. The "INTERNAL PARITY ERROR" indicator will light whenever the L-1192 hardware detects a word parity error on transfers to or from core memory.

The "INTERFACE ERROR" indicator will light whenever the L-1192 interface logic detects a character parity error on input-output data transfers. The "TIMER ERROR" indicator will light whenever the INTERVAL TIMER counts down to zero.

3.2.5.6 Operating Controls. The ON and OFF buttons control the application of power. Pressing the RUN button will cause the L-1192 to locate and execute instructions at normal operating speed until either a HALT command is decoded, the STOP button is pressed, or an internal error is detected. The STEP button will cause a single instruction to be executed each time the button is pressed. Operation of the ERROR CLEAR will reset all error toggles.

3.2.5.7 Error Controls. The "ERROR CLEAR" pushbutton resets all error indicators.

The "IGNORE ERROR" pushbutton disables the halt on instruction parity error and may be used while operating diagnostic or maintenance programs.

3.2.5.8 Power Supply Monitor and Controls. By means of the associated selector switch, output voltages of the various power supplies in the L-1192 may be displayed on the meter on the lower left-hand portion of the maintenance panel. Toggle switches are provided to control increase or decrease of power supply voltages in the diagnosis of machine malfunctions.

3.2.6 L-1192 Interval Timer

The L-1192 Interval Timer is an electronic counter operated by the master logic clock. It is composed of 32 bits (one word). It counts down by one during each clock period. It can be pre-set under program control via the I/O interface, and its current value can similarly be stored in core.

The maximum pre-set value is equivalent to approximately one hour. Resolution is approximately ten times greater than the instruction execution rate.

Section 4

L-1192 INSTRUCTIONS

The detailed descriptions of L-1192 instructions have been grouped under the following general categories:

1. Fixed Length Character Operations
2. Variable Field Operations
3. Full Word Operations
4. Program Control Operations
5. Internal Condition Operations
6. Communication Interface Operations
7. Input/Output Operations

The timing provided with each instruction includes instruction and operand access, and is given in memory cycles (each 2 microseconds). If indirect addressing is employed, one cycle must be added for each indirect level. Indexing does not increase the instruction time.

4.1 FIXED LENGTH CHARACTER OPERATIONS

These instructions provide for moving, comparing, modifying or performing arithmetic operations on single-character or fixed-length multiple character fields. These instructions are primarily used for performing character buffering functions. Most of them permit the selection of one of eight single character data registers in which the operation is to be performed. Those operations which address core memory also may specify indirect addressing, as well as indexing by one of six groups of character registers. If indirect addressing is designated, each indirect level may be independently indexed. Indexing does not normally increase the execution time, but each indirect level adds one memory cycle.

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

6	1	3	1	3	16	2
Op Code	U	R	I	X	W	C

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.

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.

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

Note: Operand 2 is usually referred to as "the specified character in memory."

1. Character Transfer

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

BBR — 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

CODING OF THE X FIELD

VALUE OF X	INDEX SELECTED
0	NO INDEXING
1	DIRECT OPERAND
2	A C — 16 BIT INDEX REGISTERS
3	B D — 16 BIT INDEX REGISTERS
4	E — 8 BIT INDEX REGISTERS
5	F — 8 BIT INDEX REGISTERS
6	G — 8 BIT INDEX REGISTERS
7	H — 8 BIT INDEX REGISTERS

CODING OF THE R FIELD

VALUE OF R	REGISTER SELECTED
0	A — 8 BIT REGISTERS
1	B — 8 BIT REGISTERS
2	C — 8 BIT REGISTERS
3	D — 8 BIT REGISTERS
4	E — 8 BIT REGISTERS
5	F — 8 BIT REGISTERS
6	G — 8 BIT REGISTERS
7	H — 8 BIT REGISTERS

Figure 4-1. Register Selection

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. Zeroes in E inhibit change in the corresponding bits of the memory location.
Timing: 2

2. Multiple-Character Transfer

The following instructions permit the simultaneous transfer of multiple-character groups between certain data registers. They are useful for rapidly loading or unloading a group of registers. The R field of these commands is utilized as 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 $\neq$ 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 $\neq$ 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 Arithmetic and Compare

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 difference in complement form.

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 $\neq$ 1, 1 IF X = 1

SUB Subtract

Subtract the character in memory from register R and put the result in register R. Underflow sets the overflow memo.
Timing: 2 IF X $\neq$ 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 register contents are less, two instructions will be skipped.
Timing: 2 IF X $\neq$ 1, 1 IF X = 1 (add one if skip occurs)

4. Logical

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 W. A skip occurs

if (1) I = 1 and all selected bits match U, or if (2) I = 0 and a mismatch exists.
Timing: 1 (add one if skip occurs)

TSP Test Parity

Test all selected bits of register R for parity. Selected bits are those which are masked by corresponding ones in W. 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 (add 1 if skip occurs)

5. Shift

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 on 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: $(N \div 4) + 1$

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: $(N \div 4) + 1$

6. Index Modify

DXR Decrement Index Register

The contents of the instruction R field (U = ∅), or the contents of the register specified by the R field is

subtracted from the index register as specified by X. If I = ∅, 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 = ∅, 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

4.2 VARIABLE FIELD OPERATIONS

These instructions facilitate the processing of variable length character fields as required, for example, by message formatting and editing. This capability is important in those applications providing message switching on a mix of both synchronous and asynchronous communication channels. It is also useful in formatting messages for magnetic tape, disc file, and/or display equipment.

Variable field instruction execution permits each of two operand addresses to specify independent field locations and lengths, up to a maximum of 16 characters. The addresses of both operands are subject to normal modification through indexing and indirect addressing. A separate register, designated the Field Operand Address Register (FOAR), is provided to hold the address and length of the first operand.

The timing for the execution of most field instructions is a function of the number of characters in each operand field (F_1 and F_2), and to some extent the locations of word boundaries within these fields. If a recomplement cycle is required, additional time is required.

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 in the following manner:

Decimal:

The four least significant bits of each character field contain the decimal digit. Bit 6 of the least significant character of each operand field contains the sign. The remaining bits will be ignored. The address and field length of operand 1 is in the Field Operand Address Register. Operand 1 is added to or subtracted from operand 2, whose length and address is carried in the instruction word. The result of the operation is stored in place of operand 2. The result of the operation is in true sign and magnitude form. If the result passes through zero, the ten's complement (recomplementing) of the result is taken. 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.

The instruction format for variable-field operations is similar to that described for the standard single character commands. U and R in the instruction word, however, specify a field length of up to 16 characters. The following instructions comprise the list of variable field operations. Timing is given in memory cycles.

IFO Initialize Field Operation

This instruction sets up the Field Operand Address Register (FOAR). U and R 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 U, R, and W/C of the specified memory location. The R field of the instruction is a part of the operation code.

Timing: 2

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: (a) No recomplement cycle: $3.25 + .75\ F_2$

(b) Recomplement cycle: $4 + F_2$

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 results of the subtraction occupies the position of the second operand. The result will carry true sign and magnitude. Underflow will cause the overflow memo to be set.

Timing: (a) No recomplement cycle: $3.25 + .75\ F_2$

(b) Recomplement cycle: $4 + F_2$

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, and carries true sign and magnitude. Overflow will set the overflow memo.

Timing: (a) No recomplement cycle: $3.25 + .75\ F_2$

(b) Recomplement cycle: $4 + F_2$

SFD Subtract Field Decimal

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

Timing: (a) No recomplement cycle: $3.25 + .75\ F_2$

(b) Recomplement cycle: $4 + F_2$

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. 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: (a) No skip: $2.5 + 5\ F_2$

(b) Skip: $3.5 + .5\ F_2$

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: (a) No skip: $2.5 + .5\ F_2$

(b) Skip: $3.5 + .5\ F_2$

MFM Move Field in Memory

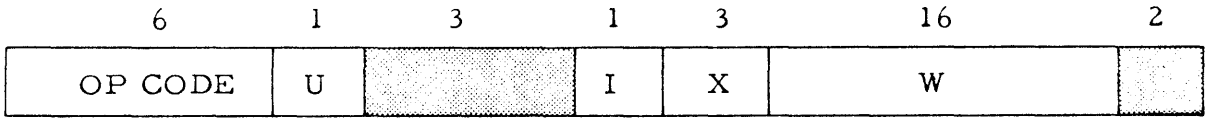
The field whose address is specified by FOAR is moved to the address specified by W/C of the instruction starting in lower memory. Reaching the end of the shorter field will terminate the operation.

Timing: $2.5 + .5\ (F_1, F_2)$ min.

4.3 FULL WORD OPERATIONS

The word instructions provide full-word (32 bit) transfer, arithmetic, and logical operations.

The format for most of these instructions is as follows:



The eight character registers A, B, ..., H are not modified by word arithmetic. Fields I, X and W are still employed to specify the location of the second operand. Load and Store word instructions use the one-bit U field to specify one of two one-word registers, R and Q. A third but non-addressable register also participates in word operations. Word operands are 32 bit numbers, with negative numbers represented in 2's complement form. The timing for the following word instructions is given in memory cycles.

WAD Word Add

Add the specified word in memory to R and put the result in memory. Overflow sets the overflow memo. Q is unchanged.
Timing: 2

WSB Word Subtract

Subtract the specified word in memory from R and put the result in memory. Underflow sets the overflow memo. Q is unchanged.
Timing: 2

WMP Word Multiply

Multiply the specified word in memory by R, and put the most significant half of the product in R and the least significant half of the product in Q. The sign of the least significant half is made the same as that of the most significant half.
Timing: 6.25

WDV Word Divide

Divide the specified word in memory into R & Q and put the quotient in Q. The remainder is placed in R in twos complement form.
Timing: 10.5

WCM Word Compare

The word in R is compared to the specified word in memory. If the register contents are algebraically greater than the word in memory the next sequential instruction is executed. If the two words are equal, one instruction is skipped. If the register contents are less than the word in memory, two instructions are skipped.
Timing: (a) No skip: 2
(b) Skip: 3

WBR Word Bring

Bring the specified word in memory to register R (U = ∅) or register Q (U = 1).
Timing: 2

WST Word Store

Store the word in register R (U = ∅) or register Q (U = 1) into the specified memory location.
Timing: 2

WCD Word Complement

Two's complement the contents of the R register.
Timing: 1

WSF Word Shift

Shift the word in register R (U = ∅) or register Q (U = 1) right (I = ∅) or left (I = 1), the number of places specified by the rightmost five bits of the instruction. Signs are not included, overflow digits are discarded. Indexing of the number of places shifted may be modified by indexing.
Timing: 1.5 + .25N

WSL Word Shift Long

Shift the combined words in register R and Q right (I = ∅) or left (I = 1), the number of places specified by the rightmost six bits of the instruction. Signs are not included, overflow digits are discarded. The number of places shifted may be modified by indexing.
Timing: 1.5 + .25N

4.4 PROGRAM CONTROL OPERATIONS

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

4.5 INTERNAL CONDITION OPERATIONS

The L-1192 has a number of internal condition indicators, which may be tested and modified.

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.
Timing: 2

System Conditions

System conditions may be set or reset, and tested for on or off condition. The available 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

TCC Test Control 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.
Timing: 1

4.6 COMMUNICATION INTERFACE OPERATIONS

Input-output character transfers for the communication channels and the slow-speed on line devices (paper tape, punched card, and typewriter/printer) provide interlaced core memory access to permit simultaneous operation.

Due to frequency of servicing the DDL 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

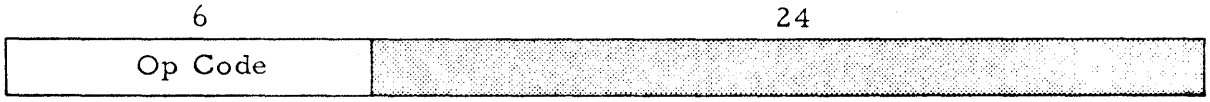
single instruction are combined the functions of testing for the presence of data in CAT 1 or CAT 2, bringing the next entry to be processed to the data registers if data is waiting to be processed, and moving hardware bookkeeping markers to the next entry to be processed.

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.

The format for the ICH and the ICL instructions is:



ICH Initiate High Priority Character Cycle

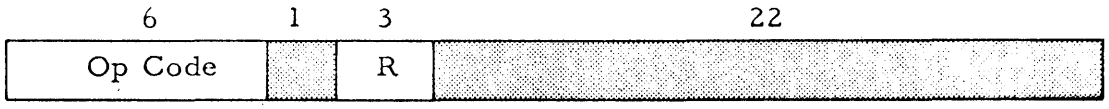
The relative positions of L1 and U1 are tested. If equal, the current instruction terminates and the next instruction in sequence is executed. If not identical, the next entry in CAT 1 is obtained from the location specified by U1. The next sequential instruction is skipped and U1 is automatically stepped.
Timing: 1 IF L1 = U1, 2 IF L1 $\neq$ 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 stepped, the following instruction is skipped, and the current instruction is terminated.
Timing: 1 IF L2 = U2, 2 IF L2 ≠ U2

The ECH and the DCH instructions control the operation of the DDL interface module scanner. The instruction format is:



ECH Enable Channel

The interface module attached to scanner position R (one of eight positions) is placed in operation. Characters will be automatically input or output under control of the Line Unit Interface module or Low Speed I/O device.
Timing: 1

DCH Disable Channel

The interface module attached to scanner position R (one of eight positions) is taken out of operation. Any fragment of a character in transmission will be lost.
Timing: 1

4.7 INPUT/OUTPUT OPERATIONS

The L-1192 Controller Processor may interface with up to sixteen I/O Channels. Each channel added to the system contains the required control logic plus a Command Register and a Block Control Register, and can communicate with up to 16 peripheral devices.

For the purpose of the following discussion, the term "instruction" refers to those operations which are decoded and executed in the instruction register. The term "command" designates those operations which are executed in the I/O interface. To activate most input/output operations, it is necessary for the L-1192 program to initiate an IIO instruction,

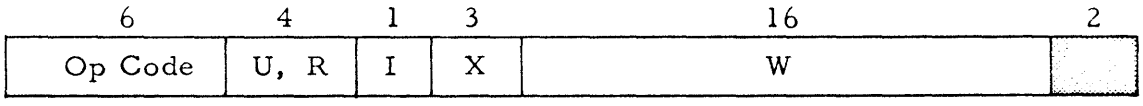
which selects one of sixteen interface channels and specifies the location of a Command Word. If the designated interface is busy, the Program Counter moves to the next instruction in sequence following the IIO. If the interface is not busy, the Command Word and a following Block Control Word are transferred to the selected I/O Interface. The Command Word, which specifies one of several available I/O commands and selects an I/O device, is transferred to the interface Command Register. The Command Word in core memory must be followed by a Block Control Word. It specifies the data transfer starting memory address and the required character count, which are loaded into the interface Block Control Register. Input/output operation then proceeds and overlaps computer instruction access and execution. The Program Counter returns the internal program to the address of the IIO instruction plus 2. A flag is set when the I/O transfer is completed, which will provide a program interrupt unless the indicator is masked.

A single interface transfer can proceed at one time on any I/O Channel. Multiple and simultaneous input-output operations therefore require more than one channel.

The parameters contained in the interface registers may be accessed by means of a Save I/O Register instruction. A Halt I/O instruction also permits the program to immediately terminate any I/O transfer operation when required. These two instructions need not be preceded or activated by an IIO instruction.

The following paragraphs provide a description and present the formats for the I/O instructions and commands.

IIO Initiate I/O Operation



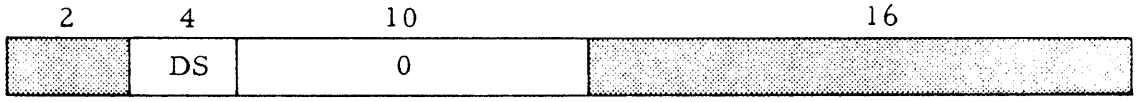
U, R These 4 bits select one of 16 Interfaces (I/O Channels) to be placed in operation, unless already busy.

I Indirect Bit

X Index specification. X = 1 has no meaning

W This address, modified by I and X specifies the core memory location of the Command Word.

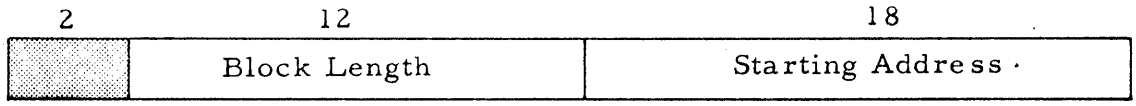
Command Word Format



DS Device Selection. This field specifies one of sixteen devices on the selected interface.

0 Device Operation Code. This field specifies the input/output command to be executed. The translation between the device operation code and the actual device function controls will be largely done by the device adapter.

Block Control Word Format



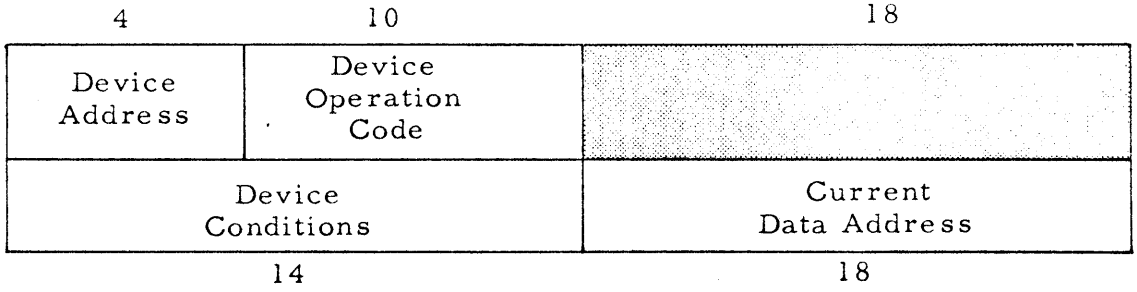
1. Starting address can be any character position in memory.

2. Block length can be as great as 4096 characters.

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



HIO Halt I/O Operation

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

Timing: 1

Section 5

PERIPHERAL EQUIPMENT

The L-1192 Controller Processor system has available a wide range of peripheral input - output equipment. Devices such as the Line Unit Console and the Card/Printer Input/Output Stations connect to the Communication Interface. The Magnetic Tape, Card Reader/Punch, Line Printer and Disc File Consoles normally attach to an I/0 Interface Channel.

5.1 LINE UNIT CONSOLE

The L-1192 Controller Processor Console, by the addition of line interface modules, may interface with up to eight simplex communication lines. The optional Line Unit Console provides the circuits for expanding this interface to as many as 64 duplex communication lines. 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 aynchronous line.

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 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 system.

5.2 MODEL 2250 SYSTEMATICS CARD/PRINTER INPUT/OUTPUT STATION

The 2250 Input/Output Station provides communication to and from the computer through the media of punched cards, page copy, and keyboard entries. The station consists of a page printer, an alphanumeric keyboard, a card read/punch, and a control module. Connections between the computer and the station I/O devices are controlled by switches on the control module. Indicator lights show the status of each device -- on or off -- and in some cases, whether the device is conditioned for input or output.

5.2.1 Operating Modes

By setting switches on the control module, the operator conditions the 2250 in any of the following nine operating modes:

Device Function	Station Operating Mode								
	1	2	3	4	5	6	7	8	9
Read Cards	X	X	X						
Accept Keyboard Entries				X	X	X			
Present Input Data to Computer	X		X			X			
Accept Output Data from Computer							X	X	X
Punch Cards					X			X	X
Print		X	X	X	X	X	X		X

Keyboard and printer communication with the computer is bi-directional when the card read and punch functions are switched off. The console operator may then make keyboard entries to the computer and receive printed replies without changing the setting of the mode switches. The 2250 exchanges data with the computer in bit-parallel character serial.

The control module transfers data to and from the computer and among the station devices on a character-for-character basis. Every character accepted as computer output is reproduced by all active station devices. Every character read from card or entered on the keyboard is transferred to all active station devices, and to the computer if it has also been selected. Other than making necessary code conversions, the station does not edit or modify data presented to it.

5.2.2 Operating Speed

The station will operate at up to 40 characters per second, input and output, when the card read and punch functions are inactive. When either card function is active, maximum speed of all station devices is 20 characters per second.

The station cannot accept output from the computer during the brief intervals when the print hammer is returning to the left margin or when the card unit is going through a feed cycle. During these intervals the station control module will present a busy signal to the computer.

5.2.3 Station Devices

1. Printer

Type:	40 character per second serial input. Printing is produced by striking a print hammer against ribbon, paper, and a rotating type drum, in that order.
Line Length:	80 characters @ 10 character per inch.

Paper Handling:	Friction feed, fanfold or continuous roll paper. The machine produces an original and three tissue copies.
Printable Characters:	Letters A through Z Digits 0 through 9 Space (blank) Special Characters - $! " - ' ? ; : & , . / ( )
Type Style:	Murray (standard teletypewriter type face); capital letters only; black printing only.
Line Feed/Carriage Return:	The print hammer is returned to the left margin by a Carriage Return code; the paper is spaced to the next line by a Line Feed code. Both codes must be keyed by the operator or presented by the computer to terminate each line of printing.

2. Alpha Numeric Keyboard

The keyboard is mounted on the front of the printer, but is not mechanically linked to the printer. Depressing a typing key produces a parallel-wire electrical output which is then routed to the printer, other active station devices, and the computer. The keyboard is locked when the printer is not selected or when the computer or card punch shows a busy signal.

Keys are provided for all printing codes and for the Line Feed and Carriage Return functions.

3. Card Read/Punch

The 2250 uses a standard IBM 24 Card Punch for card input and output. The unit operates as a punch or as a reader but not both simultaneously. When the card punch is not selected for

either reading or punching, it may be operated off-line for manual keypunching.

The card unit reads and punches 20 characters per second. Pre-determined card columns or groups of columns may be skipped at 80 characters per second. A card feed cycle requires one quarter of a second.

The card feed hopper and the card stacker each hold approximately 500 cards. If the machine runs out of cards when in punch mode, a busy signal is presented to all other active devices and to the computer. An alarm is also displayed on the 2250 control panel.

4. Control Module

The 2250 control module provides a common interface for the computer, printer, and card punch. Codes are translated as required. Compatibility of signal levels and duration is established. Data transfer among station devices and to or from the computer is enabled or inhibited as directed by switch settings on the 2250 control panel.

When the 2250 is in card read mode and the printer is enabled, a Carriage Return/Line Feed sequence is automatically generated by the control module after column 80 of each card has been read.

5.3 MAGNETIC TAPE CONSOLE

The Magnetic Tape Console contains two tape transports plus the necessary logic control and data circuits. Data is recorded at a density of 556 characters to the inch, operating at the rate of 90 inches per second. Records are separated by 3/4 inch inter-record gaps.

The end of file indication is a 3 inch gap followed by an end of file character. The 3/4 inch inter-record gap follows the end of file character. Tape can be read and recorded with either odd or even parity. Maximum tape start plus stop time is 7 milliseconds; minimum is 5 milliseconds. The unit is bi-directionally compatible with the IBM 729-IV transport.

Some degree of simultaneous operation is provided for any one I/O Interface Channel. A data transfer operation can be in process on one Tape Transport, overlapping non-data transfer operations on other Tape Transports; i.e., Rewind, Backspace Record, Backspace File, Skip Record, Skip File, Write File Mark. The use of more than one I/O channel and Tape Consoles also permits simultaneous I/O data transfers.

Tape may be moved forward or backward; however, operations requiring data transfer can take place only while tape is moving forward. Forward is from left to right, over a split head. This split head has a write station and a read station. During recording the read station checks the parity of each character 3 milliseconds after it has been recorded.

After the tape start time of 3.5 milliseconds has passed, data is transferred at the rate of 50,000 characters per second. Maximum tape stop time is 3.5 milliseconds also. Tape normally moves at the rate of 90 inches per second. There are two rewind speeds: a hi-speed 270 inches per second, and a low-speed of 90 inches per second.

5.3.1 Tape Transport On-Line Read

On-line read operation may be under "normal input" or "controlled input," as specified in the Device Operation Code in the I/O Command Word. In either case, data is read character-serial into the selected I/O Interface Data Register. When a full word has been assembled, it is transmitted through the memory access register (M) into core memory. This transfer is under control of the Block Control Register, which specifies the starting memory address and is automatically incremented for each core access.

The "normal input" mode does not utilize a character count, and the input transfer continues until an inter-record gap is reached. The "controlled input" mode is monitored by the character count in the Block Control Register, which is decremented for each input character. The transfer halts when the count equals zero.

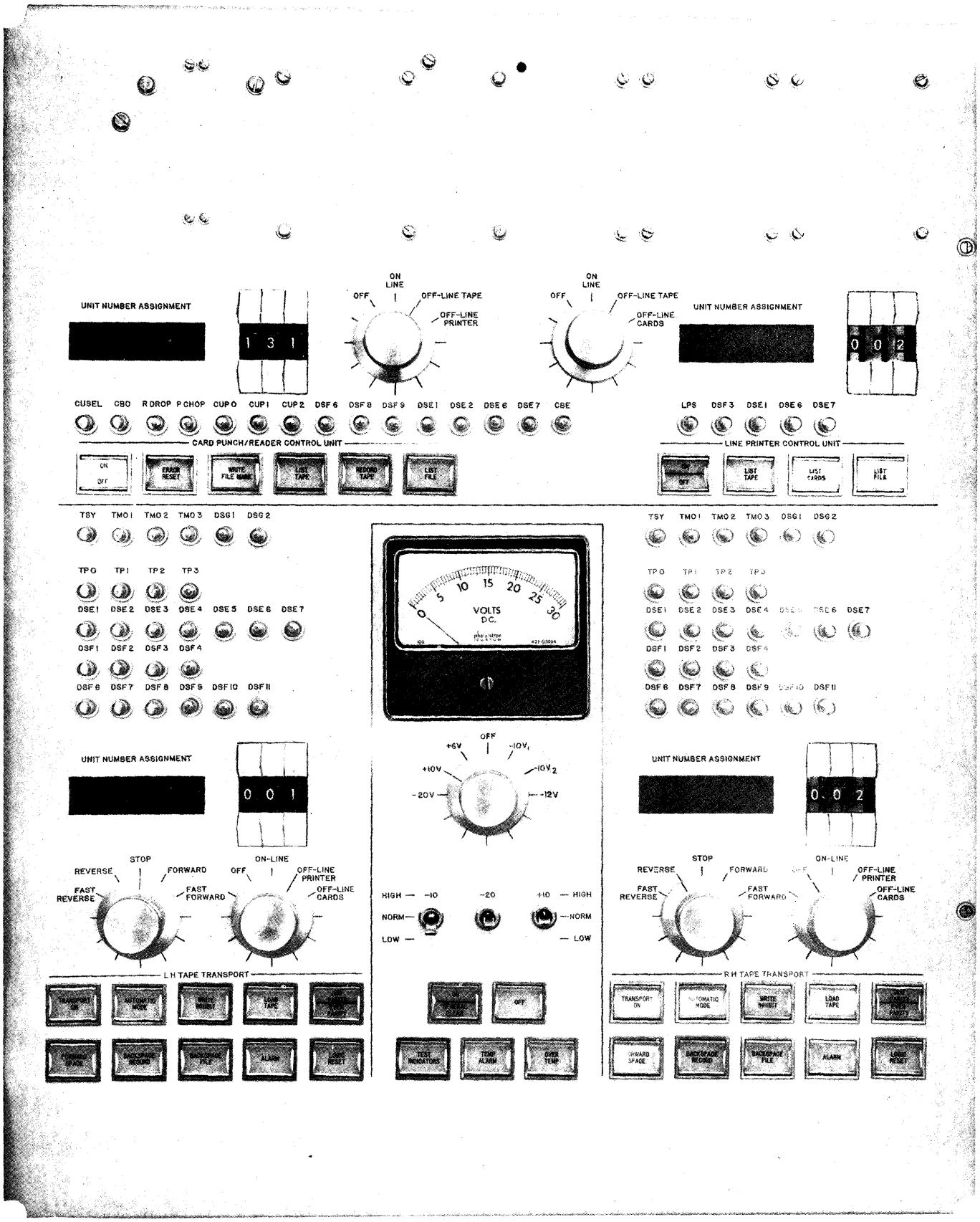


Figure 5-1. Magnetic Tape Console Control Panel

Vertical parity is checked as the tape is read into the tape console control section. At the end of each record, the horizontal check character is checked. The interface remains busy throughout the operation. When the operation is completed, the interface provide an interface not busy interrupts.

5.3.2 Tape Transport On-Line Write

The data to be written is transferred word parallel from core memory to the I/O Interface, and then is transferred character serial to the magnetic tape console for transfer onto tape. The Block Control Word specifies the starting memory location and the required character count, and the write operation continues until the count is zero. As soon as the last character is received by the magnetic tape console, the interface requests an interface not busy interrupt.

Data is parity checked first as it is read out of the core memory, then checked again upon receipt at the magnetic tape console. Three milliseconds after a character has been recorded, it is parity checked by the read side of the split head. This is called the eavesdrop check. Also a horizontal check character is recorded on tape after each block and file mark on tape. This check character is examined by the eavesdrop check, and again while reading the tape.

5.3.3 Tape Transport Off-Line Operation (Optional)

An optional off-line control unit permits direct magnetic tape to line printer or card punch, and card reader to magnetic tape or line printer operations.

The list tape operation causes the tape to be listed a record at a time on either the high speed printer (printed out) or the card reader/punch (punched cards). The record can be a maximum of 132 characters for high speed printer listing or a maximum of 80 characters for card punching. The listing process continues until an end of tape is encountered.

The list file operation causes the tape to be listed a record at a time on either the high speed printer or the card reader/punch. The listing process

continues until a file mark length or end of tape is encountered, whichever occurs first. Should end of tape be encountered first, depressing the list file button will continue the operation until a file mark is reached.

The record tape operation causes the tape to be recorded. Each record consists of the contents of one punched card as it is read and transmitted from the card reader. The recording process continues until all the cards in the card reader have been recorded.

5.4 CARD READER/PUNCH CONSOLE

The Card Reader/Punch Console (Figure 5-2) contains a Card Reader and a Card Punch mounted on the same frame. It utilizes an 80 character Core Buffer and control electronics, that are both packaged into a Card Control Unit. The equipment can read cards at the rate of 800 cards per minute, and punch at the rate of 250 cards per minute. All operations are double checked for both hardware and character validity errors in the buffer and in the unit. Both the read and punch feeds are equipped with card jam and misfeed detection devices, which can signal an error to the Central Processor. The Card Reader/Punch has five radial-type stackers, with a capacity of 1000 cards each. Two stackers are assigned exclusively to the Card Reader, and three to the Card Punch.

5.4.1 Card Reader

A Card Reader/Punch instruction, once routed to the I/O Interface, will be executed without further direct control by the Central Processor. A program interrupt is available to signal the end-of-operation. The card read instruction can transfer a specified number of characters, as designated by the Block Control Word, from consecutive locations in core memory. Eighty characters are transferred for each card read, but the transfer can be terminated by the specified character count short of a full card. There is also the option of reading until the card hopper is empty. Data flow is from the Card Reader to the Core Buffer until the buffer is full. The information is then transferred to Core Memory and the contents of the next card enter the Buffer. The effective card reading

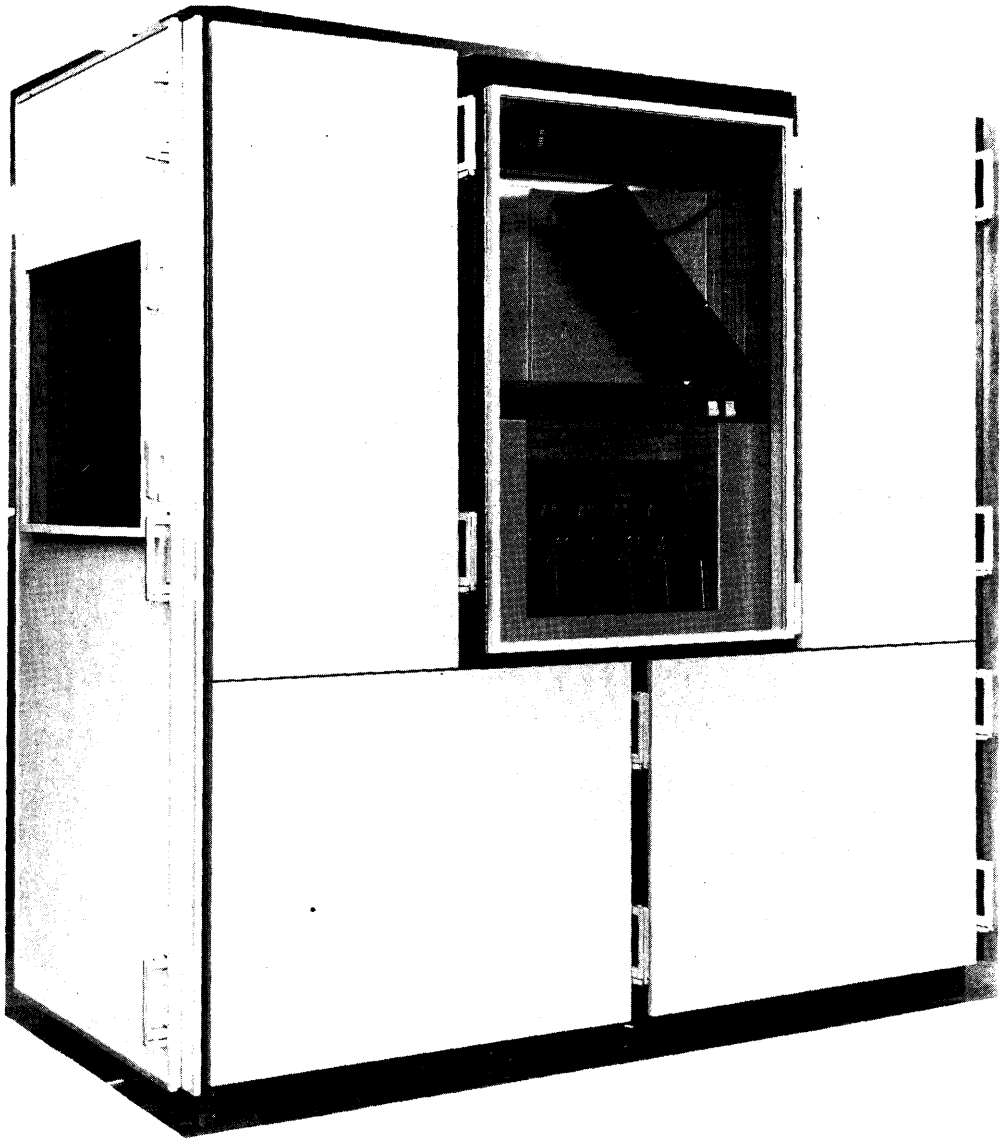


Figure 5-2. Card Reader/Punch Console

speed is 800 cards per minute, and allows a period of 74.8 milliseconds of computing between card cycles. To maintain effective card reading speed when specifying one card at a time, another card read instruction must be initiated sometime during this period.

5.4.2 Card Punch

The Card Punch instruction can transfer the specified number of characters from consecutive locations in core memory. One card is punched for each eighty characters, with any remaining columns on the last card left blank. Data flow for each card is from the Core Memory to the Core Buffer until the Buffer is loaded or the transfer is completed. The punch cycle is then initiated and, after completion, the Buffer is reloaded. The rated Card Punch speed is 250 cards per minute. The I/O Interface stays busy until the specified number of characters have been transmitted to the Core Buffer. When punching of one card at a time is specified, the I/O Interface remains busy only while data is being transferred to the Core Buffer. To maintain the rated card punching speed, however, an output punch instruction must be issued within every 240 milliseconds.

5.5 HIGH-SPEED PRINTER CONSOLE

The High Speed Printer (Figure 5-3) prints 132 characters per line at a rated speed of 1000 lines per minute. Each of the 132 print positions can print 64 characters: 26 alphabetic, 10 numbers, 27 special symbols, and a blank. The printer will handle paper from 4 to 19 inches in width, and permits copy on multiple carbon continuous form paper. Skipping and vertical formatting are accomplished through the use of a pre-punched eight channel vertical format control tape. The Line Printer can, under program control, operate in either the Controlled Vertical Format mode or in the Automatic mode. The Format mode permits the first character of the output message to select a format channel on the control tape, and the Automatic mode permits continuous line printing without vertical format or skip control. A print instruction, once routed to the I/O Interface, will be executed without further direct control by the Central Processor. A program interrupt is available to signal the end-of-

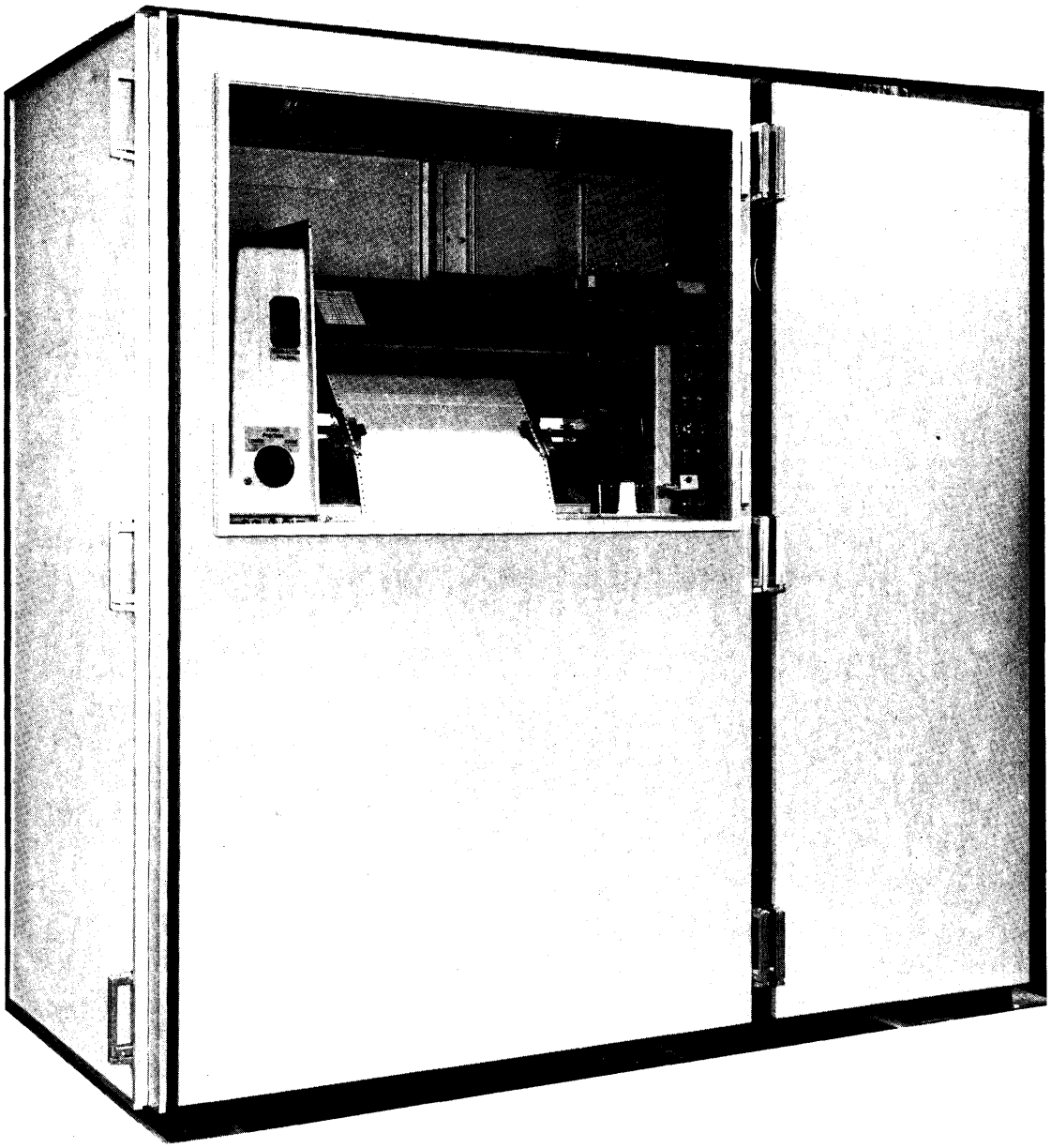


Figure 5-3. High-Speed Printer Console

operation. The output print can transfer one line, part of a line, or many lines of data from Core Memory to the Line Printer, under control of the specified character count in the Block Control Word. The Line Printer Console contains a 132-character Core Buffer. The Buffer will be loaded with the format select character and the 132 data characters from Core Memory. The print cycle is then initiated and, after completion, the buffer is reloaded. The I/O Interface will remain busy until the specified number of words have been transferred to the Core Buffer. One line will be transferred from the Central Processor at a maximum rate of one line every 59.8 milliseconds. When one line at a time operation is specified, the I/O Interface remains busy only while data is being transferred to the Printer Buffer. To maintain the rated printing speed, however, an output print instruction must be issued within every 59.8 milliseconds.

5.6 MASTER DISC FILE CONSOLE

The Master Disc File Console (Figure 5-4) provides more than 40 million data characters of storage. This capacity is divided into 324,000 blocks of information, with a block providing 128 characters. The storage is carried on six aluminum discs, which rotate on a common shaft at 900 RPM. This provides an average access time of about 35 milliseconds, and the data transfers with the Central Processor occur at a continuous rate of over 350,000 characters per second. Fixed location flying heads are utilized, with one head per track to eliminate head positioning latency and to increase file reliability. The 324,000 blocks in the File are logically divided into 240 Data Bands, each with 1350 blocks of information. A failure in a read-write head need normally only shut down one data band. An echo check is made during every disc write operation to ensure that the correct write current flows through the head for each bit recorded. This provides a check on both the write amplifier and the head. A program interrupt will alert the Central Processor in the event of an error.

The Disc File Console provides an extensive list of file commands to transfer data by fixed-address locations. An optional key search capability is also available to provide added flexibility. All file instructions,

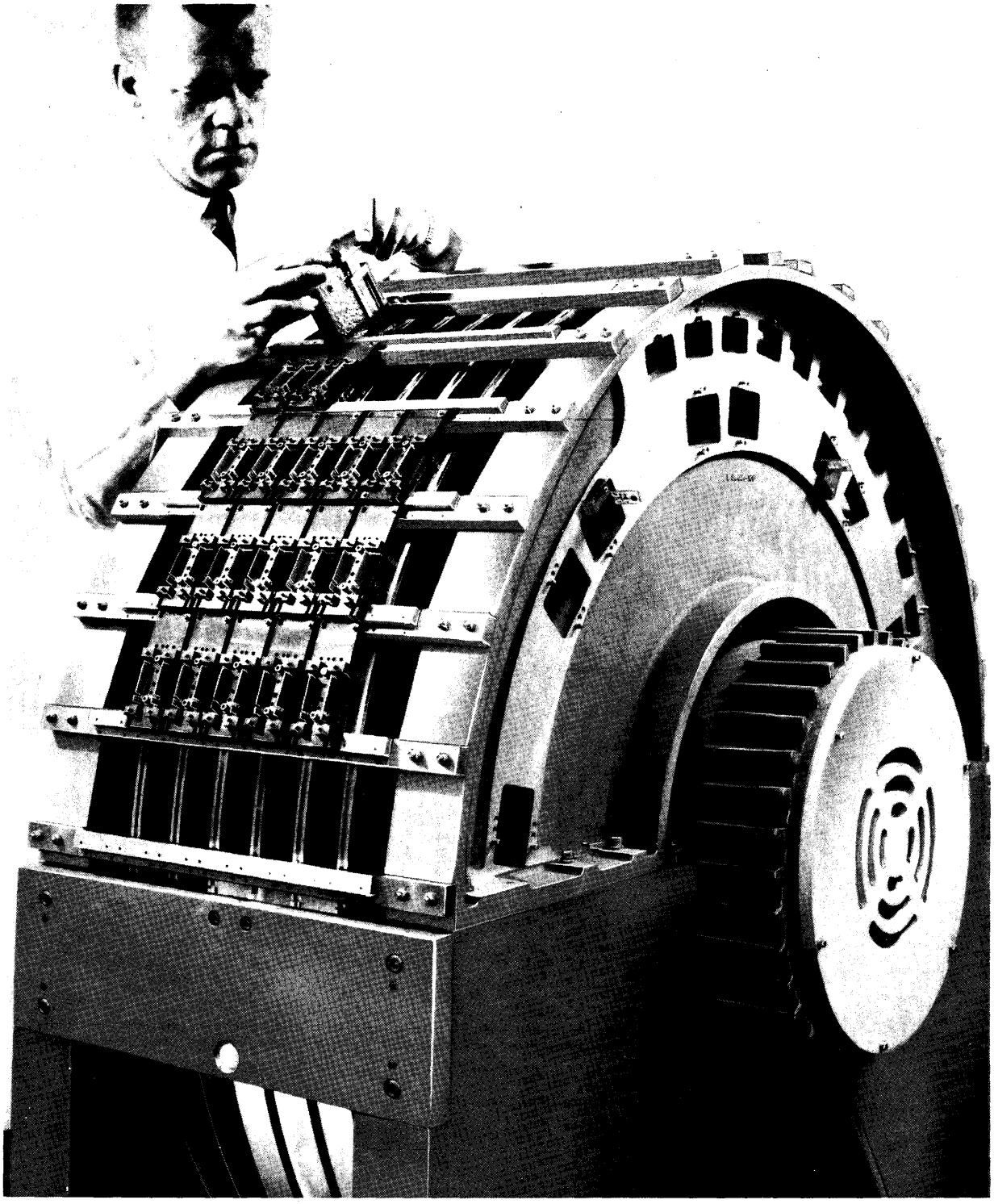


Figure 5-4. File Disc Showing Installation of Head Bar Assemblies

once accessed by the Central Processor and routed to the I/O Interface, are executed without requiring further direction or control by the Central Processor. Data access and transfers with Core Memory will overlap or interleave other interface operations, as well as internal Processor instruction access and execution. An end of Disc File data transfer will provide a program interrupt signal to the Central Processor.

The fixed address read-and write instructions specify an initial Core Memory location and a starting file block address. It may designate a transfer of from 1 to 10,000 blocks. The optional key search instructions may also retrieve up to 10,000 blocks in one search operation, with each block containing the specified key data. The criteria that may be applied in a key search are; equal to, not equal to, above a limit, below a limit, or between stated limits. Different search criteria may be applied simultaneously to the entire file block or to various specified portions of the block, in a combination of both character and bit designated keys. A search on the first 18 characters of the block will require only one disc revolution per data band, while longer keys may require additional revolutions. The file search instructions can specify either single-block or multi-block records, and may designate selected portions of each block for transfer to Core Memory. The instruction can specify that the fixed address of each block accessed by content accompany the transfer, to facilitate a subsequence return to the same file location following an update. The Processor program can test to determine whether the specified number or all matching blocks have been transferred, and retrieve a block count if necessary. It is possible to key search and simply accumulate a count on matching blocks, without requiring any transfers to Core Memory.

File instructions also permit the "obsolete" flagging of blocks that are no longer necessary, during both fixed-address read or during search access. An "Obsolete Flag Write" command may then automatically and simply enter new blocks from Core Memory into available obsolete file locations, for future access by content search, without requiring extensive internal record-keeping operations.

Section 6

SUPPORT SOFTWARE AND PROGRAMMING DESCRIPTIONS

This support software offered with the L-1192 system includes an Assembler (LAP-1192), a System Loader, a Core and Disc Dump, and Acceptance Tests. Descriptions of these programs are included in the following paragraphs.

6.1 LAP-1192 ASSEMBLER

6.1.1 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.

6.1.2 Equipment Required

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

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

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

6.1.3 Input Media

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

6.1.4 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.

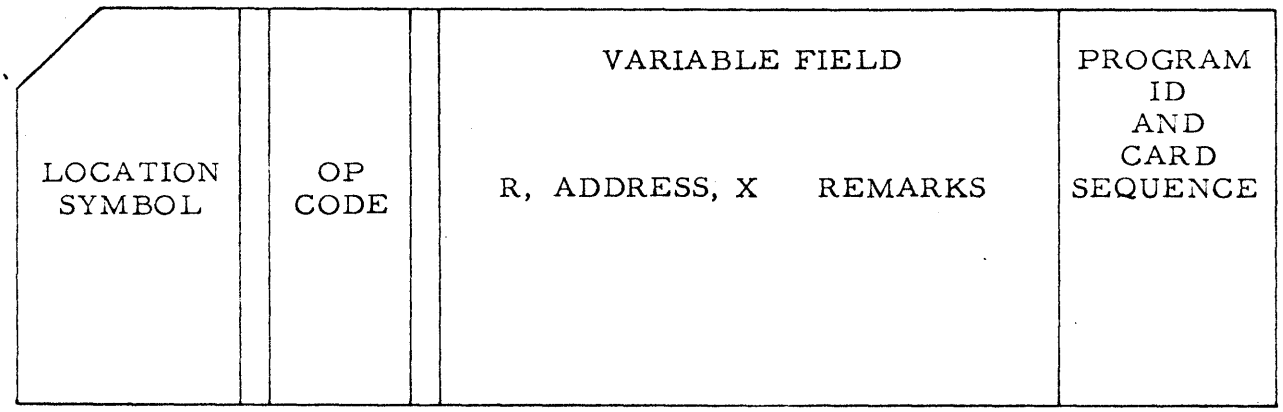
6.1.5 Features

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.

6.1.6 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.

6.1.7 Pseudo-Operations

In addition to assembling machine instructions, the LAP-1192 Assembler will also accept and process 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 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.

6.2 SYSTEM LOADER

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

6.3 CORE AND DISC DUMP

The dump program dumps specified locations of core memory or disc memory out to the console typewriter or, if available, a high speed line-printer. Data may be dumped as characters or in command format with mnemonics.

6.4 ACCEPTANCE TESTS

Acceptance tests are provided to demonstrate satisfactory performance of the L-1192 Controller-Processor System. These tests 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.

Section 7

AUTODIN PROCESSING

This section provides an example of L-1192 procedures for processing AUTODIN messages on synchronous communication lines. Figure 7-1 is an overall flow chart of the operational program.

7.1 INPUT PROCESSING (FIGURE 7-2)

The L-1192 AUTODIN program will insure that blocks of messages received from AUTODIN contain no errors before they are allowed to proceed into the main program. To aid in these error checks, the program maintains message status criteria which permit it to determine which characters will be valid for any position in a message. Blocks which contain detectable errors will be rejected. Retransmission may be requested or the line may be considered out of character frame, depending on the nature of the error.

7.1.1 Character Frame

Immediately after startup, the Line Unit will scan the incoming bit stream. No data will input to the program until the Line Unit detects an IL (idle) character. At this time characters will be permitted to proceed into the CAT 2 table, from which they will be processed by the program on a first in - first out basis. Once hardware character framing has been established, the program must count two additional contigous IL characters before logical character frame is established and message traffic can be accepted from AUTODIN. The bit scanning in the hardware can be restarted at any time by the program if it resets the CHARACTER FRAME Flip-Flop for the specified line. Character framing will be reinitiated by the program if during the establishment of logical frame a non-IL character is detected, or if an incorrect block framing character is detected in a message.

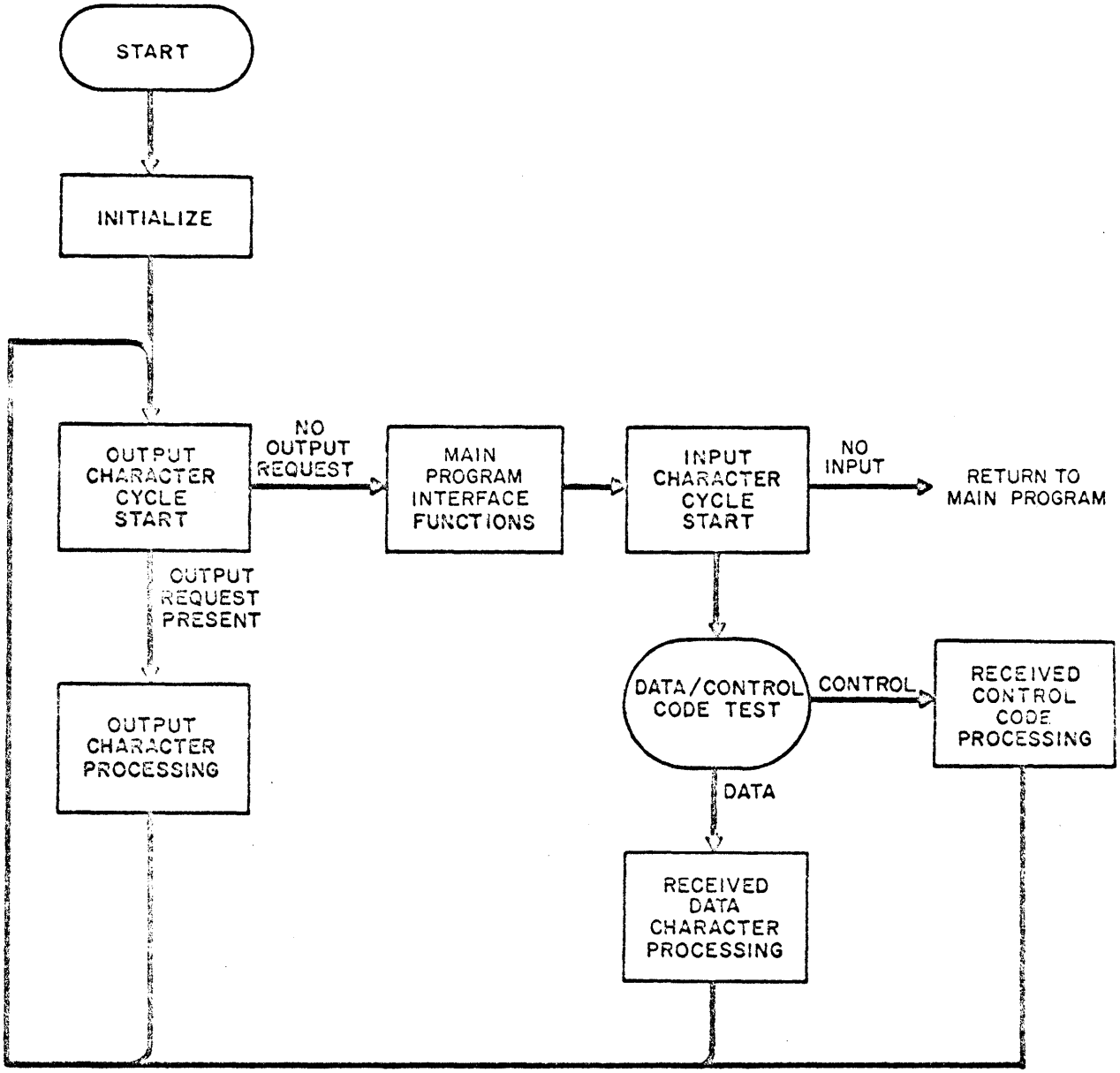


Figure 7-1. AUTODIN Program - Overall Flowchart

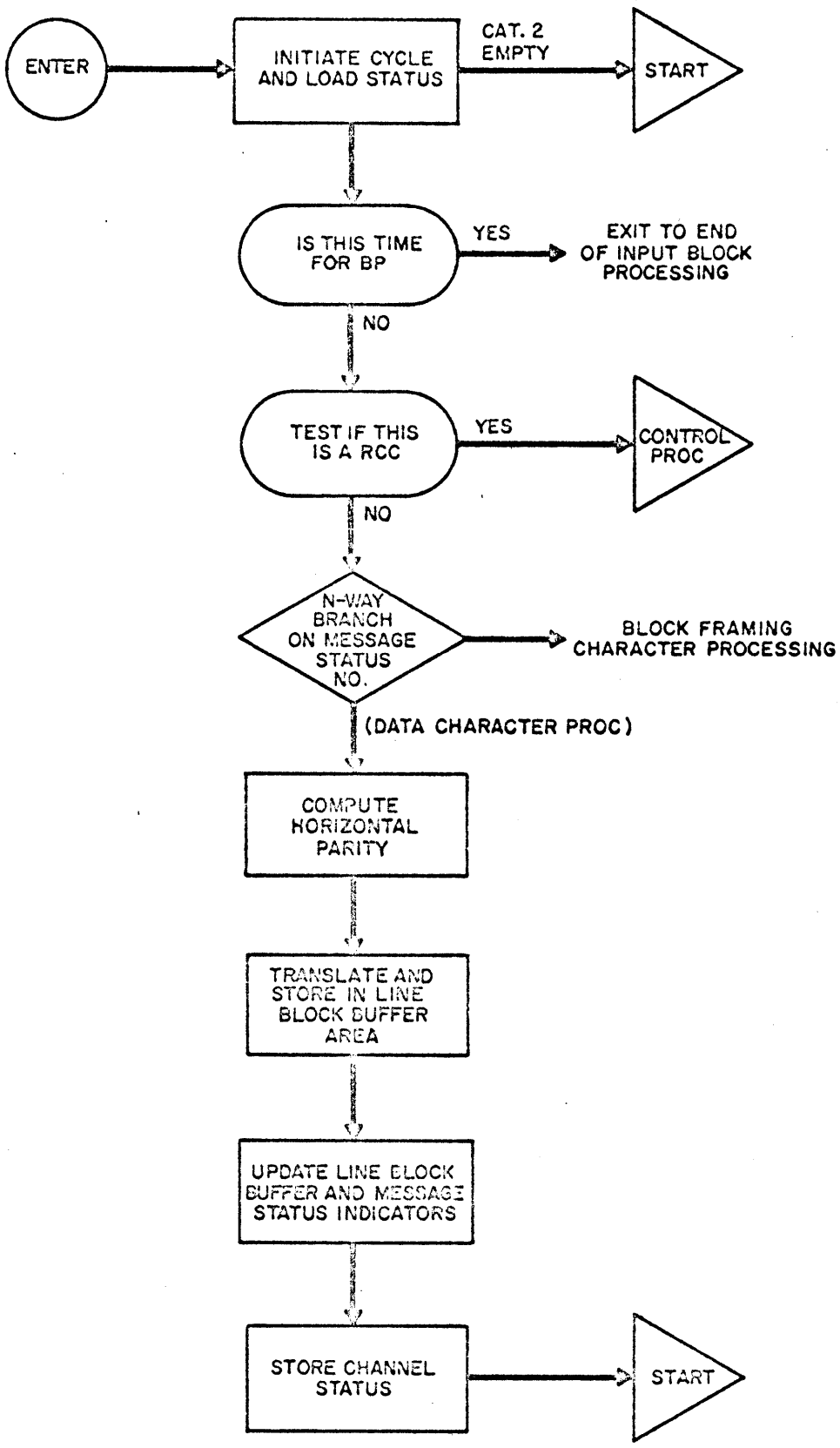


Figure 7-2. Input Character Processing

7.1.2 System States

A non-message state exists from the time the system is started up until a SOM(H or L) is detected. It also exists from the time BP is received following EOM and verified, until the SOM of the following message is detected.

A message state exists from the time a SOM is received until the message is terminated. A data state exists from the time any start of block character is received until that block is terminated. From the preceding descriptions, it can be seen that it is possible to be in a data or a non-data state while in a message state.

The following table will illustrate what constitutes a legal or illegal block framing character for any combination of data and message states:

	Message	Non-Message
DATA	EOM, EOLB after 80th data char.	Impossible State
NON-DATA	SOLB, IL	SOM, IL

From the table it can be seen that a valid Start of Block character will cause the system to go into a data state, and possibly into a message state. The receipt of a valid End of Block will cause the system to leave the data state, and possibly the message state.

7.1.3 Start of Block Processing

If the system is not in the message state, a SOM will be accepted. If in the message state but in a non-data state, a SOLB will be accepted. If any data or other block framing characters are detected in the non-data state, they are counted. When three such contiguous characters are received, the system will go out of character frame. When reframing occurs, the system will be in a non-data state. The message state will be that which existed before character frame was reset.

The acceptance of Start of Block will clear the core memory locations used for computing horizontal parity and for making up the starting and ending control fields of the block to be transferred to the Main Program. The appropriate bits in the starting control field will be set to reflect the Start of Block character received.

The following received character will be checked first for a control or answer code. If not one of these, it will be assumed to be the SEL character. The appropriate bit will be set in the starting control field if this is a TSEL message. Otherwise, the bit will remain zero. This control character will then be stored in the active line block buffering area for this line.

7.1.4 Data Characters

During the data portion of each line block, the character stream is examined for answer or control characters, which are extracted and processed. Data characters are added to the horizontal parity sum, translated, counted, and stored in the current line block buffer area. When 80 data characters have been so processed, the program will look for the block termination.

7.1.5 End of Block Operations

If the L-1192 has correctly acknowledged the previously terminated line block, the transmitter (AUTODIN) will send either EOLB or EOM, thus starting the end of block sequence. If the transmitter did not receive a correct acknowledgment it will send REP followed by IL, rather than the block termination.

Since the L-1192 program has no way of determining the current status of the transmitter, it will always accept IL after the last data character. These will not be counted or processed. Whenever the transmitter elects to send the block termination, it will be accepted. The character following EOLB or EOM will always be treated as BP. The BP received from the transmitter is compared with that computed by the program. If the two compare, and if no vertical parity errors were detected during the

receipt of the line block, it will be made available to the Main Program. (If not, the block is rejected immediately and answered with ER). If the block terminated in EOLB, the program switches to a non-data state and prepares to receive the next line block. If the block terminated in EOM, the program enters the non-message state.

7.2 OUTPUT PROCESSING (FIGURE 7-3)

Output processing involves fewer error checks than input processing, since the Main Program must control SOM, EOM, SOLB, and EOLB.

7.2.1 Character Frame

After establishing character frame by transmitting a minimum of three IL characters, the L-1192 program sets an indication in memory to signal the Main Program that the L-1192 can now accept data for that line.

7.2.2 Start of Block Processing

When a block is received from the Main Program for transmission, the first control character is examined. If either the RM or DM bits are set to 1, the remainder of the block will be ignored and the requested signal will be sent.

If the RM and DM bits are both zero, the start of block bit is examined. If zero, a SOLB character will be sent as the first character of the block. If a one, the priority bit is tested to determine if SOM-L or SOM-H will be sent. It is the function of the Main Program to insure that each message starts with SOM, ends with EOM, and contains no interspersed SOM or EOM characters.

The next time an output character request is detected in CAT1, the SEL bit of the first control character is tested. If a zero a CSEL character is sent; otherwise the message will be TSEL. On other than the first block in a message, the SEL bit will be ignored and ignore (i) will be sent as the second character of the block.

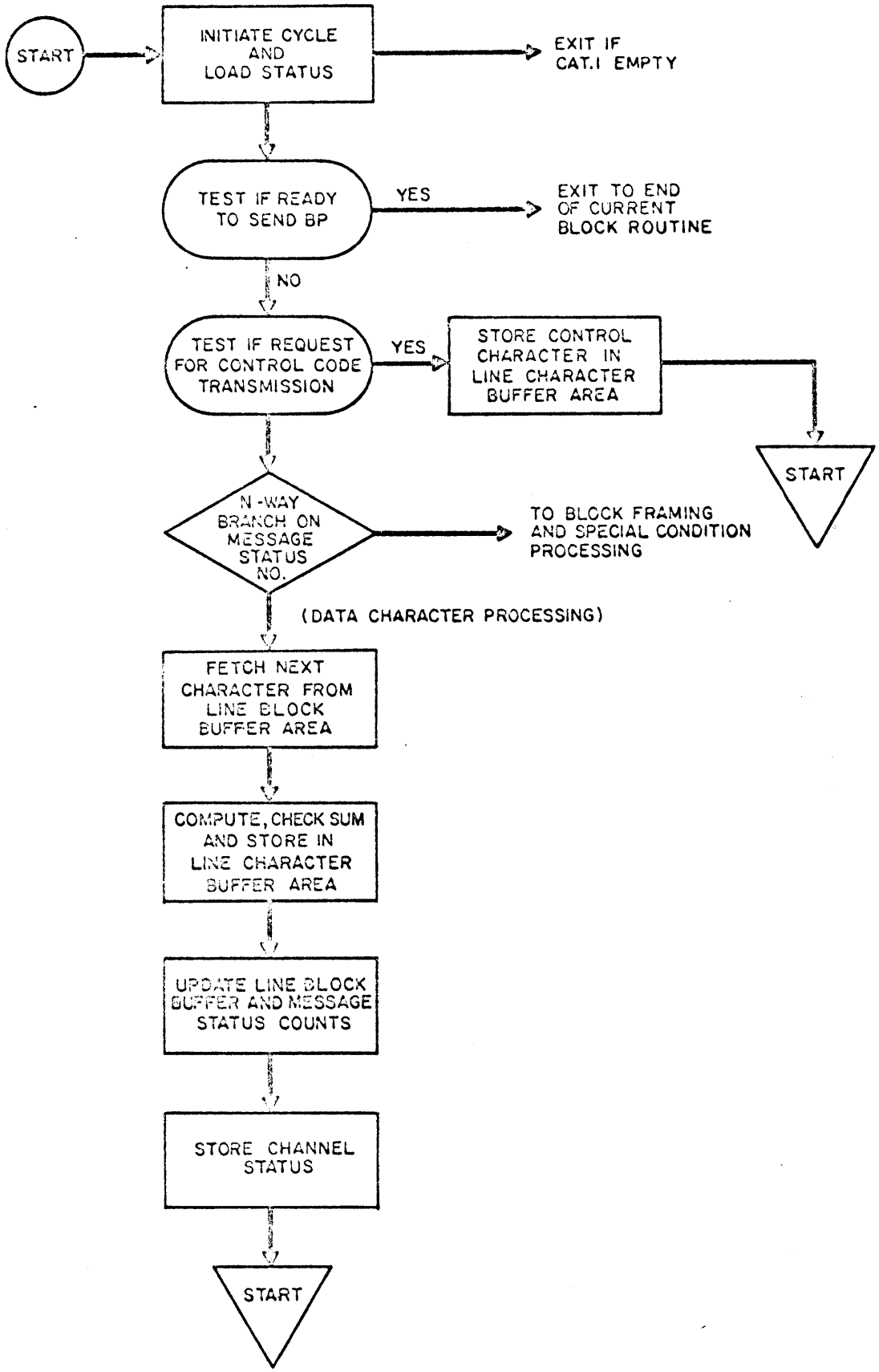


Figure 7-3. Output Character Processing

7.2.3 Data Characters

Transmission of the body of the line block commences with the third character. Each character to be transmitted is accessed from the active line block buffer area, translated to line code and stored in the line character buffer area (Output Word) for access by the hardware. After transmission each character is added to the horizontal parity sum.

7.2.4 End of Block Operations

When the 80th character has been sent, the ACK RECEIVED bit in the memo register is tested. If it is a one, the block will be terminated. If it is not a one, REP followed by IL will be sent until an acknowledgment to the previous block is received.

If the current block is to be terminated, the second control character is accessed. If the end of block bit is a one, an EOM code is transmitted; otherwise an EOLB will be sent. The EOM warning bit in the control character is stored for later use by the program. On the following character cycle, the computed BP character is output and the REPLY EXPECTED bit in the memo register is turned on.

7.3 CHANNEL COORDINATION (FIGURE 7-4)

Each transmitted line block will be acknowledged by the receiver (AUTODIN) with either the proper acknowledgment sequence (ACK1 or ACK2) if the block was correctly received, or by the ER reply if a parity error was detected and retransmission of the error block is desired.

The L-1192 program considers a reply to have been solicited by the transmission of block parity (BP). The REPLY EXPECTED bit of the memo register is set and the CORRECT ACK RECEIVED bit is reset. When any reply (ACK1, ACK2, ER, WBT) is received by the AUTODIN Program, the REPLY EXPECTED memo bit is first examined. If the reply was not expected, it is completely ignored. If the reply was expected, the following actions are performed:

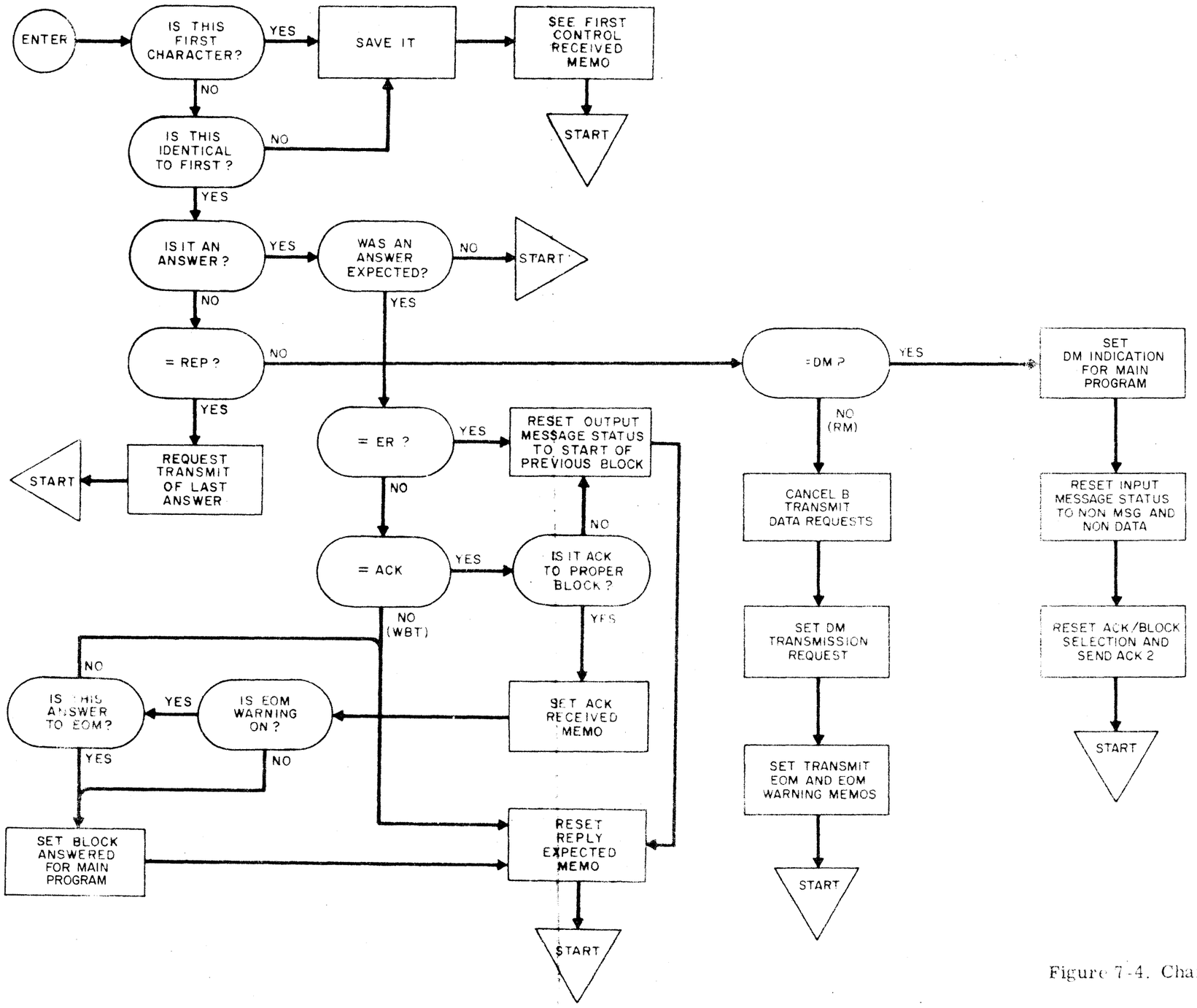


Figure 7-4. Channel Coordination Processing

1. ER or WRONG ACK

   The status information for this line is altered to cause the transmission of the current block to immediately cease, and the transmission of the previously terminated line block to be restarted. The CORRECT ACK RECEIVED indicator is set and the REPLY EXPECTED indicator is reset.

2. CORRECT ACK

   If the EOM WARNING indicator is not on, an indication in memory is set to signal the Main Program that another block for this line may be accepted. The line block status indicators in the memo register are updated to show that the just-acknowledged block area is now empty. The REPLY EXPECTED memo is reset, and the CORRECT ACK RECEIVED memo is set.

3. WBT

   Only the REPLY EXPECTED memo is reset. The CORRECT ACK RECEIVED memo remains off.

When the time is reached for terminating the following block, the CORRECT ACK RECEIVED memo is examined. If off, REP is transmitted in lieu of the block termination characters. The REPLY EXPECTED memo is set on, and 40-80 IL characters will be sent. If an acknowledgment of ER is not received during this time, the REP/IL sequence will be repeated.

7.3.1 Input Channel Coordination

Answers to line blocks are not transmitted back to AUTODIN unless they are solicited by the receipt of EOM/BP, EOLB/BP, and REP. Answers which may be sent in response to these codes include: ACK1, ACK2, ER, and WBT.

When a block is terminated, its parity is immediately checked. If any parity errors were detected, the block is answered by ER. The program purges the error line block and awaits its retransmission. If the block was received without error, the Main Program is signaled. Accepted by

the Main Program, a request is set to cause the Output Processing Routine of the AUTODIN program to return the correct acknowledgment.

Should the Main Program not accept the line block by the time the following block is to be terminated, the AUTODIN Program will send REP followed by IL as previously described. When REP is received from AUTODIN, the AUTODIN Program will immediately respond with WBT. As soon as the block has been accepted by the Main Program, the core memory cell holding the last reply sent for this line will be updated to contain the correct acknowledgment rather than WBT. When the next REP is received, the AUTODIN Program will respond with this correct acknowledgment, and normal transmission of data will resume. Under no circumstances will a received line block which has been answered by WBT be subsequently answered by ER. Likewise, no received line block which has been answered by the correct acknowledgment will be subsequently answered by WBT or ER. If a line block is answered initially by WBT, and is subsequently terminated by the sender (AUTODIN) before it has been correctly acknowledged, it will be immediately purged and character frame will be reset.

7.3.2 Processing of RM and DM

The acknowledgment of DM and RM is carried out entirely by the AUTODIN Program. Notification of receipt of these conditions is forwarded to the Main Program.

a. DM Procedure

DM will be sent by AUTODIN whenever it is desired to establish acknowledgment synchronization and/or when it is desired that the message currently being sent is to be discarded by the receiver. DM may be sent by the Main Program for the same reasons.

When DM is received from AUTODIN by the AUTODIN Program, the line block buffer area normally associated with ACK2 will be cleared to blanks. The first control character, with the DM bit

set, will be the only meaningful data transferred to the Main Program. If the alternate line block buffer area (normally associated with ACK1) contains data, it will be logically cleared by setting the associated status word to indicate empty. When the Main Program accepts the block containing the DM indicator bit, ACK2 will be sent to AUTODIN just as though a message block had been processed and normal operation will be resumed.

If the Main Program requests transmission of DM, a bit will be set in either the first or second control character. If the bit is in the first character, the AUTODIN Program will assume that the remainder of that block does not contain meaningful data. If the DM indicator bit is in the second control character, the information in the block will be sent normally. Instead of the end of block character (EOM or EOLB), however, the DM will be sent. When transferring data between the Main Program and the AUTODIN Program, the receipt of a DM request will be treated as though it were an EOM. The output portion of the AUTODIN Program will not allow any additional data from the Main Program to be accepted for that channel until the DM is acknowledged by an ACK2 sequence received from AUTODIN. When the ACK2 is received, normal operation will resume.

b. RM Procedure

RM is sent by a receiver when it is desired to reject the informating being received. RM may also be used to establish acknowledgment synchronization on the alternate side of the link, since it will be answered by DM. RM will only be accepted by a transmitter if it is in the process of sending a message.

When the receiving portion of the AUTODIN Program detects an RM, it will perform the following actions:

1. When there is no current traffic arriving from AUTODIN, one of the line block buffering areas will be cleared and a first control character inserted containing an RM bit. If there is current traffic in progress on the receive side, the

RM bit in the last control character of the next block to be sent to the Main Program will be set.

2. RM will cause the AUTODIN Program to reset the bit corresponding to the transmit side for this line in the Transfer Allow register, and will reset the line block buffer status indicators for the transmit side. This will indicate that both buffer areas are empty. An indicator will be set to indicate to the transmit side of the AUTODIN Program that an RM-DM sequence is in progress, and that no further traffic should be requested from the Main Program for that channel.

3. When the Main Program accepts the block containing the RM indicator, the receive side of the program will insert into the current transmit line block buffer area a dummy control character requesting transmission of DM.

4. The remainder of the procedure is as described for a DM initiated by the Main Program.

If the Main Program desires to reject the message it is currently receiving, the following actions will occur:

1. If an outgoing message is currently in progress, the RM bit will be set in the second control character of the next line block to be transferred by the Main Program to the AUTODIN Program. If the transmit side of the link is idle the RM bit in the first control character will be set in a dummy line block, which will be transferred to the AUTODIN Program.

2. The AUTODIN Program will forward the RM without further action. When AUTODIN responds with DM, action will be identical to that for a DM spontaneously generated by AUTODIN.

7.3.3 Establishment of Acknowledge Sync on Startup

This operation is accomplished by both 463-L and AUTODIN exchanging DM's in each direction of the data link. The generation of DM at the 463-L termination presents no problem. However, AUTODIN must be induced to send a DM by 463-L having rejected one of its messages. The following discussion describes how this may be accomplished entirely within the AUTODIN Program.

When the program is started, the initialization portion of the L-1192 program will set NO ACK SYNC indicators in both the receive and send sides of each of the line status indicator words. As the program is cycling, DM will automatically be sent on each output line to AUTODIN when character frame has been established. When the DM for each line is acknowledged, the NO ACK SYNC indicator will be reset for the transmit side of that line.

When receipt of a message is initiated on the input side for a line, the NO ACK SYNC memo is checked. If that receive link has not yet been synchronized, the transmit side will be requested to send RM. When the corresponding DM is received from AUTODIN, the NO ACK SYNC memo for that line will be reset and the ACK2 answer sent back to AUTODIN. Acknowledgment is thus synchronized in both directions for each line.

LIBRASCOPE GROUP

GENERAL PRECISION

808 WESTERN AVE. • GLENDALE 1, CALIFORNIA