INSTRUCTION MANUAL

Wang Laboratories, Inc.

836 NORTH STREET

TEWKSBURY, MASSACHUSETTS

(DP) DECIMAL 56 PRESS STROBE + L
SPC 40 " " SPC
SN 55 " " RCL L
+ 53 " " -R
C.R. 15 " " CR
L.F. 12 " " LF
— 55 " " RCL L

REFERENCE MANUAL

SERIES 4000 COMPUTER

WANG LABORATORIES, INC.
836 North Street
Tewksbury, Massachusetts 01876

September 1967

INDEX

1. INTRODUCTION

Wang Laboratories Series 4000 Computer is a true small scale general purpose computer. The basic 4000 configuration consists of an arithmetic unit (Model 4001), a keyboard (Model 4002), a memory control unit (Model 4008), a magnetic core memory unit (Model 4007), a teletype control unit (Model 4006) and a Teletype Model 33 ASR input/output terminal complete with page printer, paper tape punch, and paper tape reader.

Outstanding features of the system include 10 decimal digit accuracy on arithmetic operations and a keyboard console with 10 digit numerical display which allows operation of the computer in a desk calculator mode. Other features include random access to any memory location or storage register and facilities for decision making, looping, branching, automatic storage indexing, indirect addressing, and address modification. The various combinations of these features permit programming of the page printer for full format control, among other things.

Due to a unique parallel busline organization, Wang Series 4000 systems are easily expanded or adapted for on-line applications without modification to the basic system elements. Additional Series 4000 modules are available to interface with virtually any input or output device.

The 4000 Computer is completely self-contained and requires only a single source of 115 volt 60 cycle single-phase power. Each of the system elements, except the keyboard, is independently switched and fused and contains its own internal power supply.

2. SYSTEM ELEMENTS

2.1 Arithmetic Unit

The Model 4001 Arithmetic Unit (AU) utilizes internal circuitry which is identical to the Wang 320 electronic desk calculator. This unit gives the 4000 Computer all of the arithmetic capability of the calculator including 10 digit accuracy and single instruction simplicity for addition (+), subtraction (-), multiplication (x), division (÷), square (x^2), square root ($\sqrt{x}$), logarithmic ($\log_e x$) and exponential (e^x) operations. The arithmetic unit is completely self-contained, has its own internal timing and runs asynchronous to the memory.

The AU has four internal registers which include two independent accumulators (LA, RA) a work register (W) and a product or log register (L). Each register has a capacity of 14 four bit binary coded decimal digits plus decimal point and algebraic sign. Arithmetic operations are performed serially by digit.

The 10 most significant digits of the work (W) register are displayed with decimal point and sign on inline numerical indicator tubes. In this display, which is physically located in the keyboard module, the decimal point always appears in its proper position.

Data transfers to or from the AU always occur through the W register. Digits may be transferred to and from W either singly, as in a keyboard mode, or in blocks of 16. Block transfers are accomplished by such commands as STORE, RECALL, READ, WRITE, and TRANSFER. Each digit of the W register is also directly addressable by using sub-channel address commands in conjunction with the SINGLE DIGIT TRANSFER instruction.

Arithmetic operation codes 040 through 077 are always accepted and executed by the AU, except during transfer or address operations, regardless of whether the AU has been specifically assigned as the input or output device.

The Model 4001 Arithmetic Unit is contained in a rack mounting chassis 7" high x 19" wide x 18" deep. The unit is self-contained with its own internal power supply, is independently switched and fused and requires only 115 volt 60 cycle single-phase power. The AU is mounted in the main systems rack and contains rear panel connectors for interfacing to both the busline and the keyboard.

2.2 Keyboard

The Model 4002 Keyboard is the main operator console for the 4000 Computer. Keys are provided for every machine instruction. From the keyboard, direct access to every storage register or program instruction is possible.

A key section, which is identical to the Wang Model 320 calculator, is provided along with the standard Wang 10 digit inline numerical display. This feature permits use of the 4000 Computer as an electronic desk calculator and provides an extremely simple and versatile man-machine interface.

The keyboard is used for entering new programs (which are learned by the memory), entering data, modifying or correcting program instructions, monitoring program operation and debugging programs. In addition to the instruction keys, a set of 8 toggle switches is provided to generate any possible 8 bit busline code. Switches are also provided for initiating and controlling program operation in either a continuous or single step mode.

A lamp display is provided to monitor the status of the code buslines, allowing easy visual access to program instructions or data transmission and for checking system operations.

The Model 4002 Keyboard is a desk-top console measuring approximately 12" wide x 6" high x 14" deep. The unit draws its power from the arithmetic unit to which it connects through a single 10 foot cable.

2.3 Memory Control

The Model 4008 Memory Control (MC) package contains the logic circuitry for controlling program operation and for routing information between the buslines and the core memory.

Lamp displays are provided on the panel of this unit to monitor the memory address, the contents of the memory buffer, output buffer, address register, instruction register and the "state" generator.

The Memory Control unit is furnished in a rack mounting chassis 7" high x 19" wide x 18" deep. The unit is self-contained with its own internal power supply, is independently switched and fused and requires only 115 volt 60 cycle single-phase power. The MC is mounted in the main system rack and contains rear panel connectors which interface to the busline and the core memory.

2.4 Memory

The Model 4007 Memory package contains the magnetic core assembly with its associated drivers, sense amplifiers and buffers. The basic capacity of the Model 4007 is 1024 eight bit characters (words). This is expandable to 4096

eight bit characters by the addition of plug-in circuit boards to the existing chassis.

The memory space may be used to store either program instructions or data. Also, groups of eight adjacent memory locations may be assigned for use as storage registers. Each such register is capable of storing the entire contents of the arithmetic work (W) register, which consists of decimal point, sign and 14 decimal digits of information. Thus the 1024 character memory, if used entirely for storage, will provide 128 sixteen digit registers.

Memory addressing is in octal notation with the 1024_{10} character memory providing 2000_8 locations. Storage register addresses are also specified in octal with 128_{10} registers providing 200_8 addresses.

To simplify the process of allocating program space and storage space, the positions of the storage registers are fixed, with register zero (S0) using memory locations 1770_8 through 1777_8, register one (S1) using 1760_8 through 1767_8 and so on up to register 128 (S177) which uses locations 0000_8 through 0007_8. Since program space will normally be utilized starting with address 0000_8, this technique of assigning low numbered storage registers to high numbered memory locations allows the programmer to virtually ignore the assignment problem without fear of interference, providing that the total capacity of the memory is not exceeded. (See Appendix 1)

Allocation of the available memory space is completely arbitrary and may be randomly assigned to all programs, all storage registers, or any combination of the two. For example, if the 1024 character memory is divided equally between program and storage registers, there will result 512 instruction spaces and 64

storage registers.

Storage registers zero and one (S0, S1) are reserved for internal memory use and are not normally available to the programmer. (See Sections 5.9 and 6)

The Model 4007 Memory is contained in a rack mounting chassis 7" high x 19" wide x 18" deep. The unit is self-contained with its own internal power supply, is independently switched and fused and requires only 115 volt 60 cycle single-phase power. The memory is mounted in the main system rack and connects to the Memory Control through a single rear cable.

2.5 Teletype Control

The Model 4006 Teletype Control is the interface between the 4000 system busline and the Teletype Model 33 Automatic Send/Receive Set.

The Model 33 ASR provides the capability for punching and reading 8 level perforated paper tape as well as typing hard copy at the rate of 10 characters per second. Provision is made to control each of these operations independently by program instruction.

The Model 4006 control unit serves to convert 8 bit busline codes into standard 11 unit serial teletype signals for computer output; and to convert serial teletype signals into parallel busline codes for computer input.

The teletype unit is used to read program tapes into memory, to transmit data from tape or keyboard under program control, to punch new program tapes or data tapes and to produce typewritten computational results. Full format capability is provided by computer subroutine.

Communication between the 4006 and the 33 ASR is by standard two wire connection, allowing remote operation.

The Model 4006 Teletype Control is contained in a rack mounting chassis 7" high x 19" wide x 14" deep. The unit is self-contained with its own internal power supply, is independently switched and fused and requires only 115 volt 60 cycle single-phase power. The package is mounted in the main system rack and has rear panel connectors for the busline and the 33 ASR unit.

3. THE BUSLINE CONCEPT

3.1 General

Individual Series 4000 elements are interconnected through a set of common buslines to form a complete system. The system elements are connected in parallel to the buslines and communicate with each other through the buslines. This communication may consist of either control instructions or data transmission.

In general, one of the elements assumes control of the entire system and the other elements are assigned roles as input or output devices. The roles of input, output or control may be reassigned by control instructions issued through the bus by operator or program. Although certain elements may be designed specifically to perform only input or output functions, any element may theoretically assume control. It is also possible for particular elements to perform multiple roles.

To facilitate the issuing of control instructions, each busline element is given an address of either one or two octal digits. The elements are then referred to as channels and the addresses as channel numbers. Assignments as

input or output are made by issuing commands such as INPUT CHANNEL SELECT (ICS) or OUTPUT CHANNEL SELECT (OCS) followed by the appropriate address digit or digits. Addresses within a particular channel, such as a specific location in memory, may be addressed directly by using sub-channel control codes such as INPUT SUB-CHANNEL SELECT (ISS) or OUTPUT SUB-CHANNEL SELECT (OSS) followed by the address digits.

The flexibility of busline operation may be illustrated by considering the basic 4000 Computer system. The addresses for the elements of this system are; keyboard = 0, arithmetic = 1, memory = 7, and teletype = 6. To read a program tape from the teletype reader into memory, starting at memory location zero, the instruction sequence would be:

```
ICS
6
OCS
7
OSS
0
TRA
```

This sequence of instructions, which can be initiated by a program or by the operator pressing the appropriate keys on the keyboard, informs the system elements that the input device is the teletype (6) unit, the output device is the memory (7) unit and that the characters should be loaded into memory starting at location 0000. The information exchange would then be initiated by the TRANSFER (TRA) code and would continue until an END OF TRANSFER (EOT) code appeared on the tape. To reverse the process and punch a program tape from memory starting at location 723, the instruction sequence would be:

ICS
7
ISS
7
2
3
OCS
6
TRA

In this case, characters from memory would be transferred to the teletype unit and transmission would continue until an EOT code were encountered in memory. In similar fashion, information can be transferred from any busline element directly to any other busline element. Additional elements such as counters, voltmeters or other input and output devices, once interfaced to the busline, may communicate just as easily with the existing elements.

Busline address assignments are completely arbitrary and may be changed by switch settings internal to each element.

3.2 Busline Logic

What is loosely referred to as "the busline" is in fact a twenty-four wire cable which interconnects the system elements, all in parallel. Section 3.3 describes the function of each of the lines which is presently being used.

System elements communicate with each other by way of 8 bit binary codes. Each machine instruction is assigned a unique 8 bit code and these codes are transmitted in parallel along lines 1 through 8 of the bus cable.

To issue a command, the controlling element will impress the desired 8 bit code onto the code bus (lines 1-8) and after a short delay, will also impart a signal to the STROBE line (line 13). The strobe signal alerts all elements

that an instruction is on the code bus. One of the elements, which is conditioned to react to the particular command, will execute the required operation and signal completion by a pulse on the ACKNOWLEDGE line (line 14). Upon receipt of the acknowledgement pulse, the control element removes the instruction from the code bus, thus completing the instruction cycle.

Data transmission along the bus is accomplished in similar fashion, with the input device controlling character codes and strobe signals and the output device controlling the acknowledgement line.

3.3 Table of Common Lines

The assigned numbers correspond to the pin numbers of the busline cable connectors.

1-8 Code Bus

Lines 1 through 8 represent the code bus. The signals are DC levels, 0 volts for binary zero and -10 volts for binary one.

These lines carry control codes, or data, one character at a time. Character transmission is initiated by the STROBE signal (line 13) and in general must be ACKNOWLEDGED by a pulse on line 14 before further transmission occurs.

In octal notation the weighting for lines 1 through 8 is 1, 2, 4, 10, 20, 40, 100, 200, respectively. Thus, if code 123 were on the code bus, lines 1, 2, 5, 7 would be negative and lines 3, 4, 6, 8 would be zero.

9 Prime

The prime line, when negative, will clear the system, causing all assignments of input, output and control to be extinguished. This signal also resets critical

functions in each of the system elements so as to "prime" each element for start-up.

The prime signal may be initiated from the keyboard CLEAR ALL key and in this case will automatically establish the keyboard as the control element, for use in entering further instructions, or for use as a manual calculator.

10 Not Assigned

11 Continue

The continue line is used to single step the program when the STEP-AUTO switch is in STEP. The continue line is controlled by the STEP key located adjacent to the STEP-AUTO toggle switch.

12 Sign

The sign line carries a voltage level which indicates the sign of the arithmetic work (W) register. The level is zero volts for positive W and -10 volts for negative W. The information on the sign line is updated by the TEST SIGN OF W (TSW) command and changes only upon execution of this command regardless of the actual status of W.

13 Strobe

The strobe line carries a voltage level which indicates that a valid instruction or data character is present on the code bus. This line remains negative until the code character is acknowledged by a pulse on line 14.

14 Acknowledge

The acknowledge line is used to signal the acceptance of a code or the execution of an instruction as specified by the code bus. The acknowledge signal is a negative 10 volt pulse of approximately 10 microseconds duration.

15-20 Not Assigned

21 +11 VDC

This line is a power bus supplied by the arithmetic unit for use by the keyboard.

22 -11 VDC

This line is a power bus supplied by the arithmetic unit for use by the keyboard.

23 Chassis Ground

24 ± 0 VDC Signal Ground

4. COMPUTER INSTRUCTIONS

The coding used by the 4000 system has for convenience been chosen to match as closely as possible the ASCII convention used by the Model 33 ASR teletype. This is an eight bit code with the eighth bit used for parity. Each instruction used in the 4000 system has been assigned such a code.

In the basic 4000 system computer, the eighth or parity bit is not essential to proper functioning. This is also true for the 33 ASR teletype. That is, a code will be recognized whether the parity bit is present or not, and a parity error will have no effect on machine operation. For this reason, it is convenient here to ignore the parity bit and consider the machine codes to be of seven bits rather than eight bits.

If the seven meaningful code bits are given octal weights of 1, 2, 4, 10, 20, 40, 100, the instruction codes may be conveniently specified in octal notation. For instance, the instruction STORE has an octal code notation of 130

and this implies that bits 4, 5, and 7 are binary "ones" and that bits 1, 2, 3, and 6 are binary "zeros," since the weights of bits 4, 5 and 7 add up to exactly 130.

Using octal notation, the list of system instructions may be conveniently divided into three groups. The first group, covering codes 000 through 037, is concerned mostly with control of the teletype machine. Group two, 040 through 077, is primarily for the arithmetic unit. Group three, 100 through 137, is for system control instructions. (See Section 8.2)

The instructions are described in detail below.

4.1 Teletype Control Instructions (0-37)

Except for EOT, this entire group of instructions is devoted to the operation of the teletype machine. Instructions in this group originating on the busline will not be executed unless the teletype unit has been assigned as the output device. Instructions in this group originating from the teletype keyboard or punched tape reader will always function properly.

End of Transfer (EOT)

Code: 004

Operation: Used to terminate the transfer of information between busline elements. EOT must be the final character transmitted in the following operations: STORE, RECALL, READ, WRITE, TRANSFER.

Line Feed (LF)

Code: 012

Operation: Causes the typewriter paper feed to advance vertically by one

space.

Form Feed (Form)

Code: 014

Operation: Causes the typewriter paper feed to advance to the beginning of a new form.

Return (CR)

Code: 015

Operation: Causes the typewriter print head to return to the left most typing position. Equivalent to typewriter carriage return.

Reader On (X-ON)

Code: 021

Operation: Causes the teletype tape reader to start. Once started, the reader can only be stopped by an X-OFF code appearing on the tape or by operator intervention. The X-ON code is automatically generated by the teletype control unit when the teletype unit is assigned as the input device and a transfer operation is called for.

Punch On (Tape)

Code: 022

Operation: Activates the teletype tape punch.

Reader Off (X-OFF)

Code: 023

Operation: Stops the tape reader. This code can only be used from the tape itself. The reader will always read and transmit the next code following X-OFF on the tape before stopping.

Punch Off ("~~TAPE~~")

Code: 024

Operation: Disables the teletype tape punch.

Print On (PRINT)

Code: 030

Operation: Activates the typewriter of the teletype unit. To execute manually from the teletype keyboard, hold down the control key (CTRL) and press X.

Print Off ("~~PRINT~~")

Code: 031

Operation: Disables the typewriter of the teletype unit. To execute manually from the teletype keyboard, hold down the control key (CTRL) and press Y.

Single Step Read (SSR)

Code: 032

Operation: Causes a single character to be transmitted from the teletype reader (or keyboard). This code is automatically generated by the teletype control unit when a single digit transfer (SDT) operation is called for with the teletype assigned as the input device.

Strobe Off ("~~STROBE~~")

Code: 036

Operation: Inhibits transmission of information from the teletype to the busline by disabling the strobe signal in the teletype control unit. Useful for isolating portions of a paper tape record, such as titles and headings,

which may be of no use to the computer.

Strobe On (Strobe)

Code: 037

Operation: Enables transmission of information from the teletype to the busline. See "Strobe Off" above.

4.2 Arithmetic Instructions (40-77)

This group of instructions is devoted to the operation of the arithmetic unit. Numeric codes 60-71 are also used for addressing and data transfer. These operations will automatically be executed by the arithmetic unit regardless of whether it has been assigned as input or output.

Enter (ENT)

Code: 041

Operation: The natural log of the contents of the W register is added to the contents of the L register. Result remains in L register. W register is cleared to zero.

Result: $L = L + \ln W$; $W = 0$

$\text{Log}_e X$ (LNW)

Code: 042

Operation: The natural log of the contents of the W register is added to the contents of the L register. Result is returned to W. L is cleared to zero.

Result: $W = L + \ln W$; $L = 0$

e^X (e^W)

Code: 043

Operation: The contents of the W register are added to the contents of the L register. The antilog of the algebraic sum is returned to the W register. The L register is cleared to zero.

Result: $W = e^{W + L}$; $L = 0$

$\sqrt{x}$ $(\sqrt{w})$

Code: 044

Operation: One half of the natural log of the contents of W is added to the contents of the L register. The antilog of the sum is returned to the W register. L is cleared to zero.

Result: $W = e^{\frac{1}{2}\ln W + L}$; $L = 0$

This operation considers only the absolute value of W, and always yields a positive answer. Answer is automatically rounded off to the 10th decimal place.

x^2 (w^2)

Code: 045

Operation: Twice the natural log of the contents of the W register is added to the contents of the L register. The antilog of the sum is returned to the W register. L is cleared to zero.

Result: $W = e^{2\ln W + L}$; $L = 0$

Answer is automatically rounded off to the 10th decimal place.

$X = (X =)$

Code: 046

Operation: The log of the contents of the W register is added to the contents of the L register. The antilog of the sum is returned to the W register. L is cleared to zero.

Result: $W = e^{\ln W + L}$; $L = 0$

Answer is automatically rounded off to the 10th decimal place.

÷ = (÷ =)

Code: 047

Operation: The log of the contents of W is subtracted from the contents of L. The antilog of the difference is returned to W. L is cleared to zero.

Result: $W = e^{L - \ln W}$; $L = 0$

Answer is automatically rounded off to the 10th decimal place.

Clear Right Adder (CL R)

Code: 050

Operation: The contents of the right accumulator are set equal to zero.

Result: RA = 0

Recall Right Adder (RA→ W)

Code: 051

Operation: The contents of the W register are set equal to the contents of the right accumulator. The contents of the right accumulator are not changed. The original contents of W are lost.

Result: W = RA

Add Right (+R)

Code: 052

Operation: The contents of W are added to the contents of the right accumulator. The sum is returned to W and also remains in the right accumulator.

Result: W = RA = RA + W

(A result of zero will always have a positive sign.)

Subtract Right (-R)

Code: 053

Operation: The contents of W are subtracted from the contents of the right accumulator. The difference is returned to W and also remains in the right accumulator.

Result: W = RA = RA - W (zero always positive)

Clear Left Adder (CL L)

Code: 054

Operation: The contents of the left accumulator are set equal to zero.

Result: LA = 0

Recall Left Adder (LA→W)

Code: 055

Operation: The contents of W are set equal to the contents of the left accumulator. The contents of the left accumulator are not changed. The original contents of W are lost.

Result: W = LA

Add Left (+L)

Code: 056

Operation: The contents of W are added to the contents of the left accumulator. The sum is returned to W and also remains in the left accumulator.

Result: W = LA = LA + W (zero always positive)

Subtract Left (-L)

Code: 057

Operation: The contents of W are subtracted from the contents of the left

accumulator. The difference is returned to W and also remains in the left accumulator.

Result: W = LA = LA - W (zero always positive)

Numerical Codes

Code: 060, 061, 062, 063, 064, 065, 066, 067, 070, 071

Operation: These codes represent decimal digits 0 through 9 respectively. Any numerical code which appears on the busline, except during transfer or address operations, will automatically be indexed into the W register as if it were a keyboard entry. The original contents of W are lost. Successive digits are indexed into W until a non-digit code completes the sequence. When loading W in this fashion, DECIMAL POINT (075) CLEAR W (076) and CHANGE SIGN (077) codes are considered to be digits.

Decimal Point (.)

Code: 075

Operation: See numerical codes above.

Clear W (CL W)

Code: 076

Operation: The contents of W are set equal to zero.

Result: W = 0

Change Sign (±)

Code: 077

Operation: The sign of W is reversed. When indexing a negative number into W, the CHANGE SIGN operation must follow the digit entries, since the sign of W is always set positive by the first numerical digit.

4.3 System Control Instructions (100-137)

This group of instructions is used to control the flow of information within the computer. In the examples below, system elements have the following addresses; 0 - Keyboard, 1 - Arithmetic Unit, 6 - Teletype, 7 - Memory.

Continue (CTU)

Code: 103

Operation: Used to complete the transfer of control or temporary control from one element to another, or to return control when ending a temporary control mode. (see examples below)

Temporary Transfer of Control (TTC)

Code: 104

Operation: Used to establish the temporary control mode. Must be followed by CCS and the address of the desired channel, plus sub-channel address if necessary. Actual entry into TTC mode is caused by CTU code which follows addressing.

Example:

```
TTC
CCS
0
CTU
```

This program sequence issues temporary control to channel 0 which is the keyboard. The program stops and the keyboard is activated. Control reverts to the program when the operator presses the CTU key on the keyboard.

Control Channel Select (CCS)

Code: 110

Operation: Used to transfer control to another element. Must be followed by address digits and, if necessary, the sub-channel address. Actual transfer of control is caused by CTU code which follows addressing.

Example:

```
CCS
7
CSS
2
3
CTU
```

This sequence of instructions, issued from the keyboard, will transfer control to channel 7, which is the memory unit. The first instruction read from memory will come from memory location 23.

Control Sub-Channel Select (CSS)

Code: 111

Operation: Used to specify the location from which the next program instruction should be taken. This instruction may be used by the operator to start a program from a particular memory location (see example above) or may be used within a program to cause branching from one portion to another portion of memory.

Input Channel Select (ICS)

Code: 112

Operation: Used to specify which element is to be the input device. Must be followed by appropriate address digits. Actual selection occurs when address is terminated by any subsequent non-digit code. Previously selected input, if any, is disabled by this command.

Example:

```
ICS
6
CL R
```

This sequence of instructions will select channel 6, which is the teletype, as the input device.

Input Sub-Channel Select (ISS)

Code: 113

Operation: Used to select an address within a given input channel, such as one input to a multi-channel scanner or a specific memory location.

Example:

```
ICS
7
ISS
1
2
3
TRA
```

This sequence selects channel 7, the memory, as input and specifies further that information starting at location 123 should be transferred to the busline.

Output Channel Select (OCS)

Code: 114

Operation: Used to specify which element is to be the output device. Must be followed by appropriate address digits. Actual selection occurs when address is terminated by any subsequent non-digit code. A previously selected output, if any, is disabled by this command.

Example:

```
OCS
6
CL R
```

This sequence selects the teletype (6) unit as the output device.

Output Sub-Channel Select (OSS)

Code: 115

Operation: Used to select an address within a given output channel such as a specific memory address.

Example:

```
ICS
0
OCS
7
OSS
0
TRA
```

This sequence specifies that the keyboard (0) is input, that the memory (7) is output, and that subsequent codes transferred from the keyboard should be loaded into memory starting at location zero.

Return (RTN)

Code: 116

Operation: Used to return from a sub-routine to the main program. Causes the next program instruction to be taken from an address which had previously been stored as a result of a JMP operation. See JUMP below.

Jump (JMP)

Code: 117

Operation: Causes the program to branch to an address which is specified by digits which follow the JMP code. The program remembers where the jump originates by saving the address of the command which follows the jump sequence. Use of the RTN command (see above) then allows continuation of the main program as if the jump had not occurred. Useful in branching to sub-routines.

Example:

Address	Code	Address	Code
100	JMP	753	-
101	7	.	-
102	5	.	-
103	3	.	RTN
104	x^2		
	+R		

Sequence starting at 100 causes program to branch to location 753, while saving address 104. Sub-routine starting at 753 ends with RTN code, causing program to resume at step 104.

Single Digit Transfer (SDT)

Code: 120

Operation: Causes a single character to be transferred from the assigned input device to the assigned output device. The transferred character may be any eight bit code. If the transmitted character happens to be a machine instruction, the instruction will not be executed.

Example:

```
ICS
7
ISS
1
0
2
5
OCS
6
SDT
```

This sequence causes the character in location 1025 of memory (7) to be sent to the teletype (6).

Transfer (TRA)

Code: 121

Operation: Causes a block of characters to be transferred from the input device to the output device. The last character of the block must always be EOT. The transmitted characters may be any eight bit codes. If any of the transmitted codes are machine instructions, they will not be executed.

Example:

Address	Code	Address	Code
0	ICS	23	R
1	7	24	E
2	ISS	25	J
3	2	26	E
4	3	27	C
5	OCS	30	T
6	6	31	EOT
7	TRA		

The sequence of instructions starting at memory location zero causes the message "REJECT" to be typed on the teletype unit.

Load W (LDW)

Code: 122

Operation: Used to terminate an address sequence when the next required instruction must be a digit entry into W.

Example:

```
OCS
6
LDW
2
√W
```

The LDW instruction serves to isolate the address digit 6 from the W register entry 2. If LDW were not used, the system would attempt to select

channel 62 as output.

Write (WRT)

Code: 123

Operation: Causes the entire contents of the arithmetic W register to be transferred to the assigned output device. Exactly seventeen characters are transferred, which include 16 decimal digits followed by the EOT code. The format is fixed, with the first digit representing decimal point position, the second digit representing sign (zero for plus, nine for minus) and the remaining 14 digits specifying the value of W. The contents of W are not changed by this operation.

Example:

```
LDW
1
2
3
.
4
OCS
6
WRT
```

This instruction sequence will cause the teletype to print 30123400000000000. (EOT is a non-printing code but it would appear on tape if the punch were activated.)

Read (RD)

Code: 124

Operation: Causes a transfer of information from the assigned input device into the arithmetic W register. The number of digits transferred may vary but the first digit must be decimal point location and the second digit must be

sign. Any number of numerical codes may then follow and the final code must be EOT. The original contents of W are lost.

Example:

```
ICS
6
READ
```

This sequence will read data from the teletype tape or keyboard into the arithmetic W register.

Test Sign of W (TSW)

Code: 125

Operation: This instruction is used in conjunction with the JMP and CSS commands to perform conditional branching and looping. Conditions of the test are such that the operation following the TSW command will be performed if the sign of W is negative or will be skipped if the sign of W is positive.

Example:

```
-
TSW
JMP
1
2
3
CLR
```

If W is negative, the program branches to instruction 123, otherwise it continues in sequence.

Store (STR)

Code: 130

Operation: Causes the entire contents of the arithmetic W register to be transferred to a storage register in memory. If the command is followed by address digits, those digits specify the desired register. If no address

digits follow, the information will be stored in a predetermined location as described in "Load Storage Address (LSA)" below. The address, if used, must be terminated by a non-digit code.

Example:

```
LDW
1
2
.
3
STR
2
5
CL R
```

This sequence causes the contents of W (+12.300000000000) to be stored in register 25.

Recall (RCL)

Code: 131

Operation: Causes the contents of a particular storage register to be returned to the arithmetic W register. If the recall command is followed by address digits, those digits specify the desired register. If no address digits follow, the information will be recalled from a predetermined register as described in "Load Recall Address (LRA)" below. The address, if used, must be terminated by a non-digit code.

Example:

```
RCL
2
5
CL R
```

This sequence returns the contents of register 25 to W. The contents of register 25 remain unchanged. The original contents of W are lost.

Load Storage Address (LSA)

Code: 132

Operation: This instruction sets the address which is used in conjunction with STORE commands which have no associated address digits. The address digits which follow the command are retained in memory, used as required, and automatically indexed after each usage.

Example:

```
LSA
 2
LDW
 1
STR
LDW
 5
STR
CL R
```

This sequence causes +1.0000000000000 to be stored in register 2 and +5.0000000000000 to be stored in register 3. Subsequent store commands not having address digits will cause numbers to be stored in successive registers. Store commands with addresses are handled normally and do not affect the "LSA" address.

Load Recall Address (LRA)

Code: 133

Operation: This instruction is followed by address digits which are retained in memory and used in conjunction with RECALL commands which have no associated address digits.

Example:

```
LRA
 1
 0
```

```
RCL
CL R
+R
RCL
 3
CL L
+L
RCL
+R
```

This sequence leaves the contents of register 3 in the left accumulator and the sum of the contents of registers 10 and 11 in the right accumulator.

5. PROGRAMMING

5.1 Memory Organization

Program instructions are stored in the core memory. They are loaded either manually by using the keyboard, or automatically by using the tape reader on the teletype. As the program runs, the instructions are read sequentially from memory and executed.

The basic 4000 Computer has a memory capacity of 1024 instructions. Each of the 1024 memory locations has an address which is specified in octal notation. The octal memory addresses therefore range from 0000_8 to 1777_8.

The first program instruction will generally be located at address 0000_8 and will be followed in order by the remaining instructions. The program instructions will be processed sequentially from the first to the last, although certain decision making capabilities enable the program to branch from one list of instructions to another, and return, if necessary, or to perform a certain group of instructions repetitively.

The available memory space may also be used as storage registers. Each

storage register occupies memory space equivalent to eight characters (or program instructions), and is capable of storing the entire contents of the arithmetic W register. The storage registers are also addressed in octal notation. The 1024 step memory, if used entirely as storage registers, will provide 128_{10} registers or 177_8 storage addresses.

Storage registers physically occupy memory space starting at address 1777_8 and working backwards toward 0000. Thus, storage register zero (S0) occupies memory spaces 1770_8 through 1777. Similarly, S1 occupies 1760_8 - 1767_8, S2 occupies 1750_8 - 1757_8 and so on (See Appendix 1)

Since the program instructions will normally start at location 0000_8 and work towards 1777_8, and the storage registers start at 1777_8 and work back towards 0000, the programmer need not worrry about interference of program space and storage space, providing that the total capacity of the memory is not exceeded. The total space used will equal the number of program steps plus eight times the number of registers, and must be less than 1024 total (all counts taken in decimal notation).

Storage registers zero (S0) and one (S1) are used internally by the computer and are not normally available to the programmer. The information stored in this area includes the program counter, return address, read address, write address, store address and recall address. This information is accessible, and ways of using it are discussed below in Section 6.

5.2 Manual Data Entry

Quite often it is desirable to use a program which permits manual entry of

variables from the keyboard. However, during the running of a program, the series 4000 keyboard is normally disabled to avoid accidental interference by the operator. Thus, when keyboard entries are required by the program, a sequence of instructions is used which will issue temporary system control to the keyboard and stop the program. This instruction sequence is:

```
TTC
CCS
 0
CTU
```

The operator keys in the required data and restarts the program (i.e., returns control to memory) by pressing the CTU key.

5.3 Arithmetic Expressions

Arithmetic expressions are programmed according to the basic rules of algebra. The general procedure is to write the instructions in the same order that would be used to perform the required operations manually from the keyboard. The example below illustrates a program to solve the equation $C = \sqrt{a^2 + b^2}$.

The program starts at step 0, runs to step 3 and stops after transferring temporary control to the keyboard. The operator keys in variable "a" and re-starts the program by pressing the CTU key. The program runs to step 12 and stops again. The operator keys in variable "b" and presses CTU. The program completes the calculation, then jumps back to step 0 and proceeds down to step 3 where it stops as before. Here the operator can read the result "c" on the keyboard display and the program is ready to accept a new input "a".

No.	Command	Comment
00	TTC	
01	CCS	
02	0	
03	CTU	INPUT a
04	X^2	a^2
05	CLR	
06	+R	a^2
07	TTC	
10	CCS	
11	0	
12	CTU	INPUT b
13	X^2	b^2
14	+R	$a^2 + b^2$
15	$\sqrt{}$	$\sqrt{a^2 + b^2}$
16	JMP	
17	0	
20	EOT	
21		

Example 5.3.1

Arithmetic Operations

5.4 Storage Registers

The use of storage registers is very straightforward. Suppose that in the preceding example (Section 5.3), variable "a" was to be found in register 6 and variable "b" was to be found in register 15. Suppose further that the the result $C = \sqrt{a^2 + b^2}$ was to be stored in register 23 for later use. The program sequence to accomplish this is shown below:

No.	Command	Comment
40	RCL	
41	6	a
42	x^2	a^2
43	CL R	
44	+R	a^2
45	RCL	
46	1	
47	5	b
50	x^2	b^2
51	+R	a^2+b^2
52	$\sqrt{x}$	C
53	STR	
54	2	
55	3	
56	CL L	
57		
60		
61		

Example 5.4.1

Use of Storage Registers

Registers are simply assigned as needed, generally starting with S2 since S0, S1 are reserved for internal use by the computer. Storage addresses are in octal notation, and care must be taken to terminate each address by a non-digit instruction as shown in the example.

5.5 Decisions, Looping And Branching

There are two types of program loop. The first type, which is an endless loop, is illustrated in the example of Section 5.3. An endless loop uses the

JMP or CSS command to unconditionally recycle through a series of instructions. Once started in an endless loop, a program will recycle until changed by the operator.

A second type of loop involves decision making instructions which allow the program to exit from the loop under certain circumstances. This same decision making capability is used for branching operations.

All decisions in the 4000 Computer are based on a test of the sign of the arithmetic W register. By suitable programming, this single test can implement the general decision problem of whether a given number is less than, equal to, or greater than another number. The instruction used for this purpose is the TEST SIGN OF W (TSW) command.

The TSW command is usually followed by a JMP or CSS operation. If the sign of W is negative when TSW is executed, the program will do the following operation (JMP or CSS) as expected. However, if the sign of W is positive when TSW is executed, the program will ignore the following operation (JMP or CSS) and proceed on to the next instruction. Any address digits associated with the JMP or CSS instructions are considered to be part of the operation, and will also be ignored.

Example 5.5.1 below illustrates the general decision problem. Given two numbers A and B, this program will branch to instruction 100 if B is greater than A; will branch to instruction 200 if B is less than A; or will proceed to instruction 23 if B equals A. This program takes advantage of the fact that zero will always be positive following an arithmetic operation.

No.	Command	Comment
00	RCL	
01	3	A
02	CL R	
03	+R	A
04	RCL	
05	4	B
06	-R	A-B
07	TSW	if W is
10	JMP	negative
11	1	go to 100
12	0	B>A
13	0	
14	CL R	
15	-R	B-A
16	TSW	if W is
17	JMP	negative
20	2	go to 200
21	0	B<A
22	0	
23	CL L	B=A
24		
25		
26		
27		

Example 5.5.1

Conditional Branching

Example 5.5.2 below illustrates branching, conditional looping and the use of a sub-routine. Two separate but related programs are shown in the example. The first program, starting at instruction 00 will calculate N! Where N is entered from the keyboard. The second program, starting at instruction 13, calculates 1/(A + 2)! Where A is entered from the keyboard. Both programs use a common sub-routine which calculates N! The sub-routine performs (N-1) loops before returning to the main program.

Notice that the loop in the sub-routine must use the CSS command rather than the JMP command so that the return address to the main program will not be lost.

No.	Command	Comment	No.	Command	Comment
00	TTC		40	CL R	
01	CCS		41	+R	N
02	0		42	RA→W	
03	CTU	INPUT N	43	ENT	
04	JMP		44	1	
05	4		45	−R	
06	0		46	CL L	
07	JMP		47	−L	
10	0		50	1	
11	EOT		51	+L	
12			52	TSW	
13	TTC		53	CSS	
14	CCS		54	4	
15	0		55	2	
16	CTU	INPUT A	56	LDW	
17	CL R		57	1	
20	+R	A	60	X=	N!
21	2		61	RTN	
22	+R	A+2	62		
23	JMP		63		
24	4		64		
25	0		65		
26	÷=	1/(A+2)!	66		
27	JMP		67		
30	1		70		
31	3		71		
32	EOT		72		
33			73		
34			74		
35			75		
36			76		
37			77		

Example 5.5.2

5.6 Storage Indexing

The method for accessing storage registers directly by address was covered in Section 5.4 A second method will be described here which permits indirect access to registers and allows for automatic indexing.

Suppose, for instance, that it is desirable to store a list of twenty numbers, which are to be entered from the keyboard. Suppose further, that the first number is to be stored in register 2 and that successive numbers are to be stored in successive registers. The procedure to accomplish this is to specify the starting point using the LSA command, then to issue store commands without addresses. The computer will recognize that the address is missing, look up the prearranged address, use it, index it, and return it to memory for further use. These prearranged addresses, the so-called "store address" and " recall address" are held in the reserved portion of memory.

Example 5.6.1 below illustrates a routine which accepts twenty keyboard entries, stores them in successive registers starting at S2 and automatically proceeds on with the program after the twentieth entry.

Example 5.6.2 below illustrates a routine which will multiply the contents of registers 10 through 17 by the contents of register 6, replacing the original contents of each register by the respective product.

No.	Command	Comment
00	LSA	
01	2	
02	CL R	
03	2	
04	0	
05	-R	
06	TTC	
07	CCS	
10	0	
11	CTU	KB ENTRY
12	STR	
13	LDW	
14	1	
15	+R	
16	TSW	
17	JMP	
20	6	
21	CL L	
22	- - -	
23		
24		
25		
26		
27		

Example 5.6.1

Storage Indexing

No.	Command	Comment
40	LRA	
41	1	
42	0	
43	LSA	
44	1	
45	0	
46	CL R	
47	3	
50	-R	
51	RCL	
52	ENT	
53	RCL	
54	6	
55	X=	
56	STR	
57	LDW	
60	1	
61	+R	
62	TSW	
63	JMP	
64	5	
65	1	
66	CL L	
67	- - -	

Example 5.6.2

Store and Recall Indexing

5.7 Input-Output Control

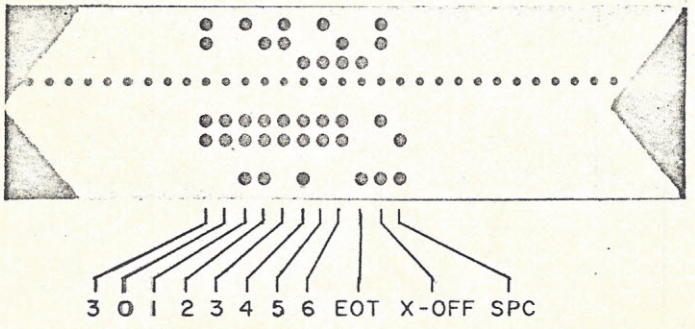
I/O control programming is very straightforward. If the program requires data input from a particular device, instructions are issued to assign that device as the input channel. If that input itself has a number of inputs or sub-channels, as in the case of a scanner, then sub-channel address commands are also used.

To direct the input data into the arithmetic unit, the READ command is used. In this case, the data must have a fixed format as described in Section 4.3

To direct the input data to another unit, such as the memory, the second unit must be addressed as output; using sub-channel addressing if required.

Once assignments of input and output are made, they remain in force until a new assignment is made. (The CLEAR ALL key also disables input-output assignments.) Information is transmitted from input to output using the TRA or SDT commands, or from input to the arithmetic unit using READ or from the arithmetic unit to the output using WRITE.

Suppose the teletype reader contained a data tape which held the number +123.456, the tape would be punched as follows:



For purpose of illustration, example 5.7.1 shows a routine which will read the tape into memory (7), loading it into locations 100 through 110, then read the data from memory (7) into the arithmetic (1), then write the data from the arithmetic (1) back out to the teletype (6). The space occupied by the data in memory is also shown. Notice that the EOT is the last code registered in memory as a result of the transfer.

No.	Command	Comment	No.	Command	Comment
40	ICS	}	100	3	
41	6	TT	01	0	
42	OCS	TAPE	02	1	
43	7	TO	03	2	
44	OSS	} MEMORY	04	3	
45	1		05	4	
46	0		06	5	
47	0		07	6	
50	TRA		110	EOT	
51	ICS	}	11		
52	7	MEMORY	12		
53	ISS	TO	13		
54	1	A.U.	14		
55	0	}	15		
56	0		16		
57	RD		17		
60	OCS	} A.U.	120		
61	6	} TO	21		
62	WRT	} TT	22		
63	...		23		

Example 5.7.1

Input-Output Control

Information may also be transmitted one character at a time by using the SDT command. This is illustrated in the next section.

5.8 Formatting

A general formatting sub-routine is available for the 4000. This discussion is intended to illustrate the mechanics of formatting typewritten data on the teletype.

In the example below is a sub-routine which prepares a specific format of the form xx.xxx, with two digits before and three digits after the decimal point. Upon entering the sub-routine, the data to be typed is in the arithmetic W register.

The first portion of the program, steps 0 → 11 serve to align the decimal point and round off the number after the third decimal place. This operation in effect moves the five digits to be typed to the right end of the W display, automatically rounding off due to the x = operation.

Steps 12 → 20 then transfer the entire W word into memory, filling locations 100 → 120. At this point, the five digits to be typed are known to occupy spaces 7 → 13.

Steps 21 → 33 add "space" and "EOT" codes into 114, 115, respectively.

Steps 34 → 43 cause the first two digits to be typed, using the SDT command.

Steps 44 → 47 type the decimal point.

Steps 50 → 54 transfer the remainder of the word to the teletype.

No.	Command	Comment	No.	Command	Comment	No.	Command	Comment
00	ENT		40		FIRST	100		dec of
01	*		41		TWO	01		sign
02	0	ALIGN	42	SDT	DIGITS	02		1
03	0	AND	43	SDT		03		2
04	0	ROUND	44	ISS	STEP	04		3
05	0	OFF	45	5	OUT	05		4
06	0		46	6	DECIMAL	06		5
07	0		47	SDT	POINT	07	X	6
10	1		50	ISS	TRANSFER	110	X	7
11	X=		51	1	REST	11	X	8
12	OCS		52	1	OF	12	X	9
13	7	TRANSFER	53	1	WORD	13	X	10
14	OSS	TO	54	TRA		14	SPC	11
15	1	MEMORY	55	RTN	RETURN	15	EOT	12
16	0		56	DP		16		13
17	0		57	SPC		17		14
20	WRT		60	EOT		120		EOT
21	OSS		61			21		
22	1	ADD	62			22		
23	1	SPC, EOT	63			23		
24	4	TO	64			24		
25	ICS	ASSEMBLED	65			25		
26	7	WORD	66			26		
27	ISS		67			27		
30	0		70			130		
31	5		71			31		
32	7		72			32		
33	TRA		73			33		
34	OCS		74			34		
35	6	STEP	75			35		
36	ISS	OUT	76			36		
37			77			37		

Example 5.8.1

Formatting

5.9 Reserved Storage

Storage registers zero and one are reserved for internal machine use. In these are kept the program address, return address, write address, read address, recall address, and store address. These addresses are used by the machine as follows:

5.9.1 Program Address

A program counter is used to address program instructions in sequence. This same counter is also used to control transfer operations and store - recall operations. When one of these diversions interrupts the normal counting sequence of the program counter, the program address is saved in reserved storage. Then when the required operation is complete, the program address is retrieved from storage, loaded back into the program counter, and the program continues.

5.9.2 Return Address

When a JMP operation is executed, the address of the next instruction following the JMP operation is saved as the "Return Address". The program counter then branches to the address specified by the JMP command. If a RTN command is then executed, the "Return Address" is retrieved, loaded into the program counter, and the program resumes.

5.9.3 Write Address

The "Write Address" is the address used by memory when executing a TRA operation as output. It is loaded by an OSS operation when memory is output. It is automatically updated following a transfer.

5.9.4 Read Address

The "Read Address" is the address used by memory when executing a TRA operation as input. It is loaded by an ISS operation when memory is input. It is automatically updated following a transfer.

5.9.5 Recall Address

The "Recall Address" is used by memory when a "Recall" operation is executed without a fixed address. The "Recall Address" is set by the LRA command, and is automatically indexed after each use.

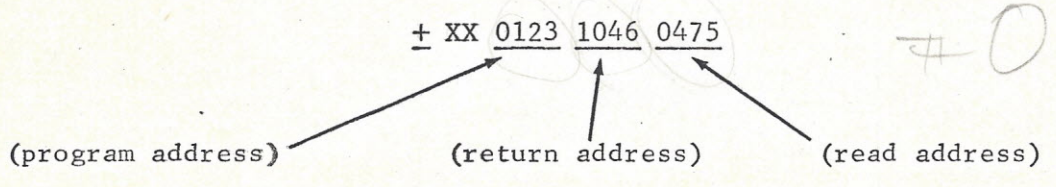
5.9.6 Store Address

The "Store Address" is used by memory when a "Store" operation is executed without a fixed address. The "Store Address" is set by a LSA command and is automatically indexed after each use.

6. USING RESERVED STORAGE

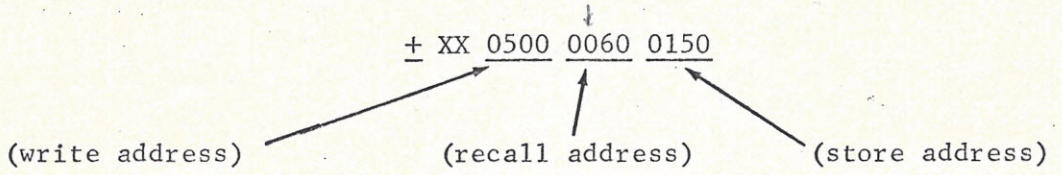
The addresses held in reserved storage are all in octal notation. They are carefully assembled so that they may be recalled into the W register and used by the programmer.

Specifically, register 0 holds the "Program Address," "Return Address," and "Read address." If the program address were 123, the return address 1046, and the read address 475, then following a recall of register zero, the W register would have the following contents.



The sign, decimal point, and first two digits are meaningless but they may be set, if desired, by a normal store operation, and will not be changed by the normal operation of the machine.

Similarly, register 1 holds the write address, recall address and store address. If these were 0500, 6, 15, respectively, then following a recall of register one, the contents of W would be:



(Notice that the least significant digit in the store and recall address is always zero, i.e., that the address is shifted relative to all other addresses.)

The programmer is free to use reserved storage as desired, but care must be taken with register zero so as not to upset the program address.

Example 6.0.1 below illustrates a technique for manipulating register one in order to index two sets of recall operations simultaneously. This is done by using two recall addresses, holding the current one in register 1 and the alternate in register 2 or 3. By swapping the contents of 1, 2, and 3, it is possible to index two recall lists. This technique may of course be extended. The example illustrates summing two rows of a matrix and depositing the sums in a third row. Each row contains eight terms.

No.	Command	Comment	No.	Command	Comment	No.	Command	Comment	No.	Command	Comment
00	CL L	LOOP	40	2	SWAP	00			40		
01	8	8	41	RCL	AGAIN	01			41		
02	-L		42	3		02			42		
03	LRA	SET	43	STR		03			43		
04	1	RECALL	44	1		04			44		
05	0	ADDESS,	45	STR	a_{3j}	05			45		
06	RCL	MOVE	46	LDW		06			46		
07	1	TO	47	1	DECREMENT	07			47		
10	STR	S2	50	+L	AND	10			50		
11	2		51	TSW	TEST,	11			51		
12	LRA	SET 2ND	52	JMP	LOOP	12			52		
13	2	RECALL	53	2	8 TIMES	13			53		
14	0	ADDRESS	54	0		14			54		
15	LSA	STORE	55	CL L		15			55		
16	3	ADDRESS	56	. . .		16			56		
17	0		57			17			57		
20	RCL	a_{1j}	60			20			60		
21	CL R		61			21			61		
22	+R	a_{2j}	62			22			62		
23	RCL		63			23			63		
24	1	SWAP	64			24			64		
25	STR	RECALL	65			25			65		
26	3	ADDRESSES	66			26			66		
27	RCL		67			27			67		
30	2		70			30			70		
31	STR		71			31			71		
32	1		72			32			72		
33	RCL	a_{2j}	73			33			73		
34	+R	$a_{1j} + a_{2j}$	74			34			74		
35	RCL		75			35			75		
36	1		76			36			76		
37	STR		77			37			77		

Example 6.0.1

Use of Reserved Storage

7. OPERATING PROCEDURES

7.1 Interconnections And Turn-On Procedures

The system components should be interconnected as shown in Figure 7.1. Notice that each chassis in the system rack has its own power switch and fuse, and that each plugs into the main rack power strip. The main power cord is connected to a 115 volt, 60 cycle, single-phase power outlet. Power must be off while connections are being made.

Once all connections are made and power is turned on, there is no reason why the electronics should not be left on continuously. The power required by the system circuitry is less than that required by a few household lamps, and there are virtually no parts which will wear out. The teletype machine should, however, be turned off when not in use. (See Section 7.2)

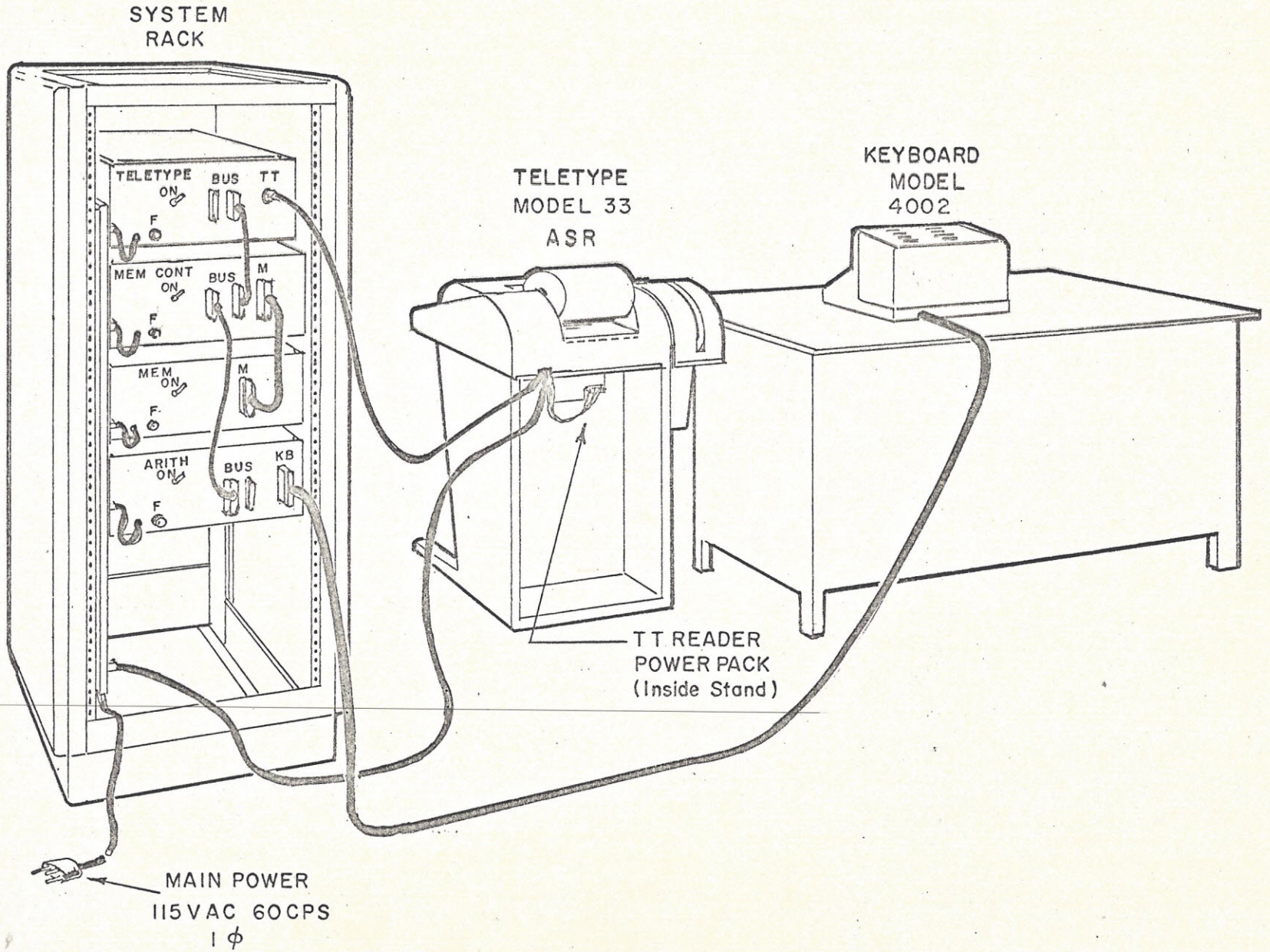
Each of the system chassis has a power indicator lamp on its front panel which indicates that the internal logic supply (-12 volts) is operating. Individual fuse ratings are stencilled above the fuse holders on the rear panel.

The keyboard receives its power from the arithmetic unit, to which it connects. (See Section 7.3)

Each of the chassis may be turned on or off without affecting the other chassis.

7.2 Teletype Controls

The Teletype Model 33 ASR console is shown in Figure 7.2.0. The power switch is located at the lower right front corner of the unit and is labelled LINE-OFF-LOCAL. When the machine is not in use, the power switch should be left in the



SERIES 4000 COMPUTER INTERCONNECTION DIAGRAM FIG. 7.1

OFF position. For operation with the computer, the power switch should be left in the LINE position. To use the teletype independently, the power switch should be set to LOCAL.

7.2.1 Punch Controls

Four control buttons are located on the punch unit, just in front of the paper tape reel.

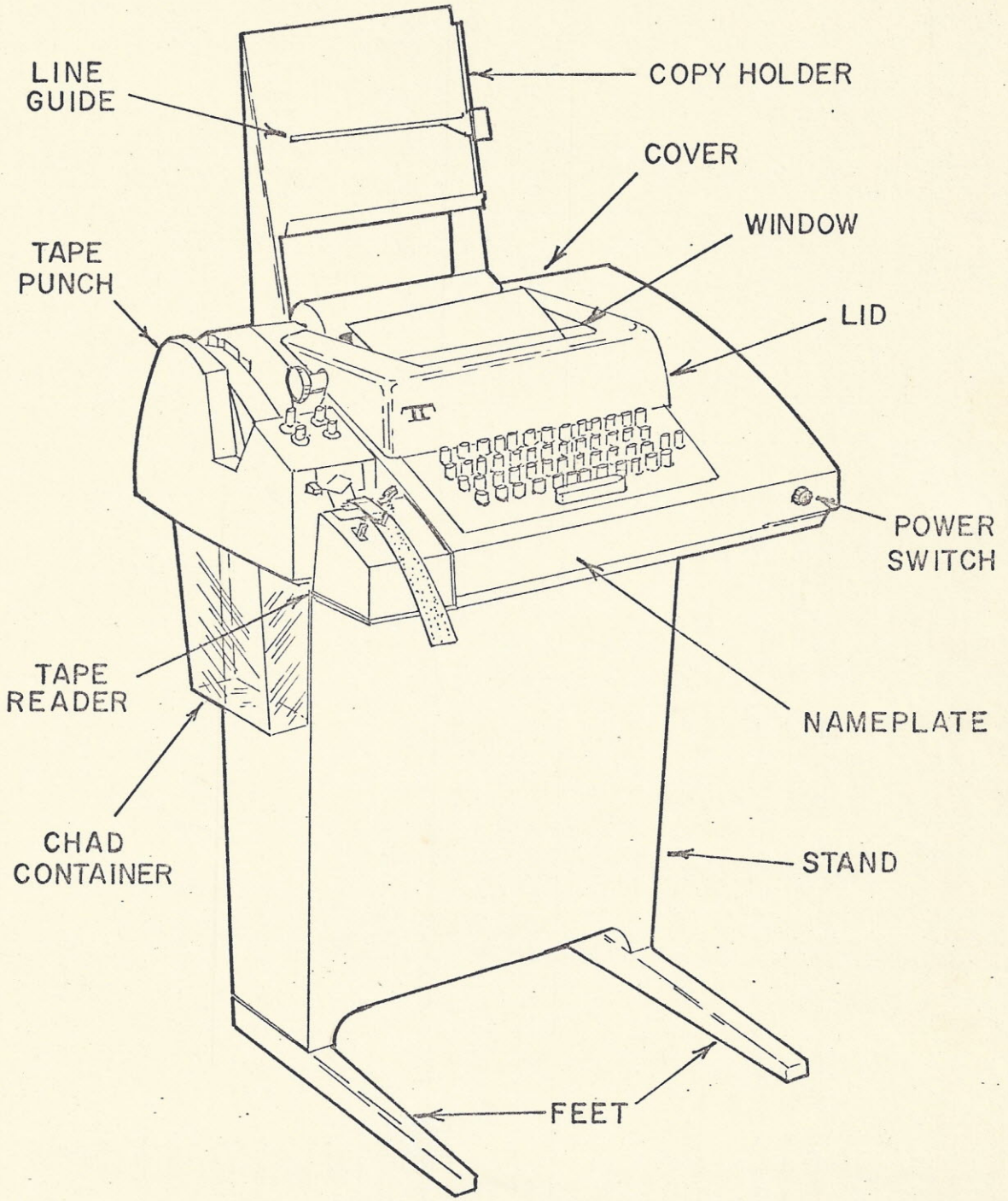
7.2.1.1 The REL pushbutton disengages the paper advance mechanism, allowing easy threading or removal of the paper tape.

7.2.1.2 The B. SP. pushbutton backspaces the tape by one character allowing the character just punched to be deleted by use of the RUB OUT key.

7.2.1.3 The LOCK ON and UNLOCK pushbuttons control operation of the punch, in association with the TAPE and ~~TAPE~~ keys. With the UNLOCK button engaged, punching may be initiated by holding down the CTRL key and pressing the TAPE key. To inhibit punching, press CTRL and ~~TAPE~~. The TAPE and ~~TAPE~~ commands may of course also be issued by the computer.

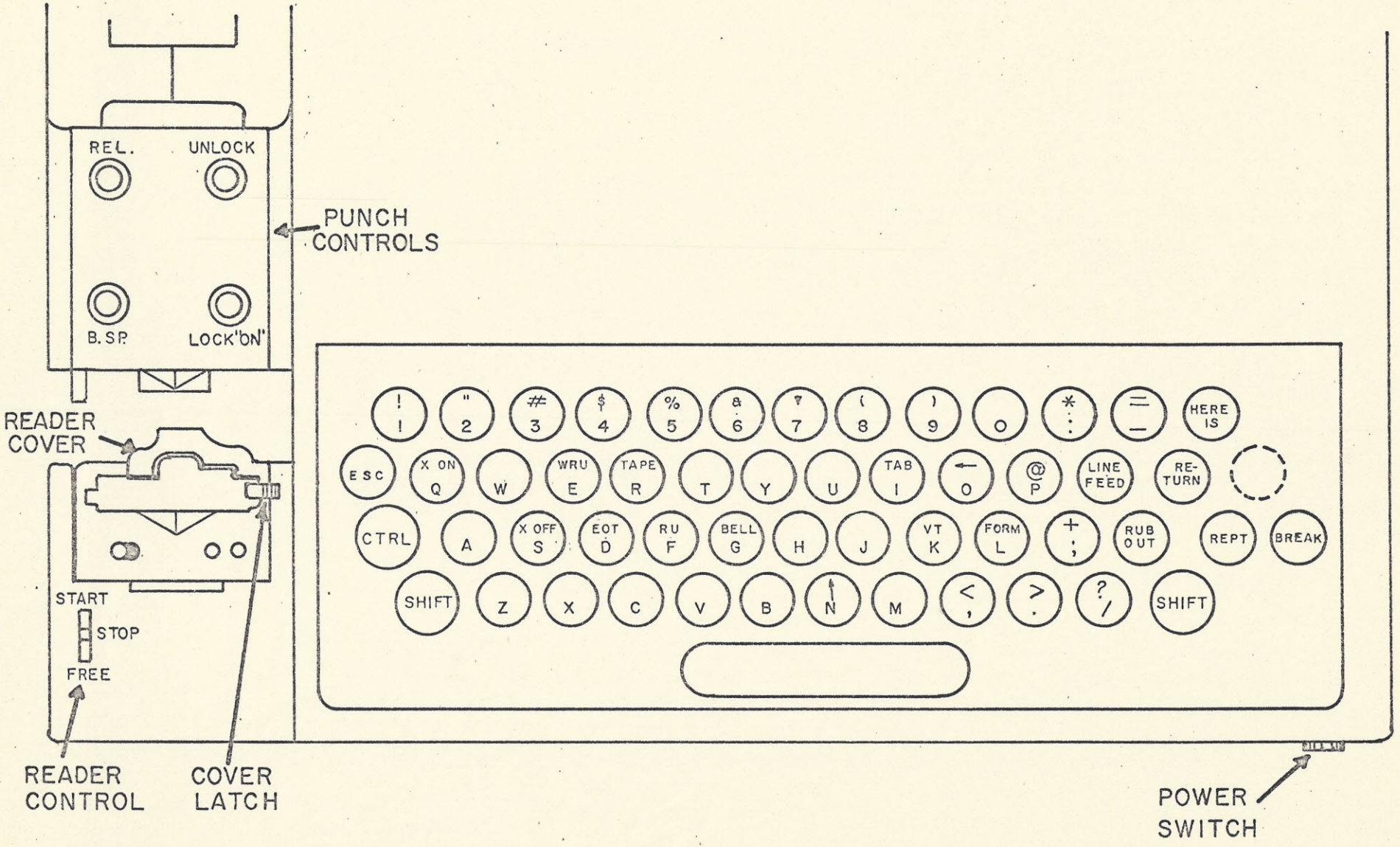
7.2.2 Reader Control

The reader is located directly in front of the punch unit. Located on the reader is a three-position switch labelled START-STOP-FREE. This switch is normally left in the STOP position, even when reading is to be controlled by the computer. With the switch on STOP, reading may be initiated by holding down the CTRL key and pressing X-ON. Once the reader has started, it can only be stopped by an X-OFF code appearing on the tape, or by moving the reader switch to FREE, or by lifting the reader cover. The X-ON command can of course be issued by the computer.



TELETYPE MODEL 33 ASR CONSOLE

FIGURE 7.2.0



TELETYPE CONTROL FIGURE 7.2.1

The FREE position disengages the reader mechanism, allowing the tape to be pulled freely through the reader.

The START position manually forces the reader to operate, and is useful for duplicating tapes.

The reader is interlocked so that it will only operate when tape is present. To load the reader, release the clear plastic lid covering the read station by moving the latch at the right. Place the tape in the reader, being sure that the sprocket teeth engage the tape feed holes, and close the cover. The tape advances through the reader from the rear towards the front.

7.2.3 Teletype Keyboard

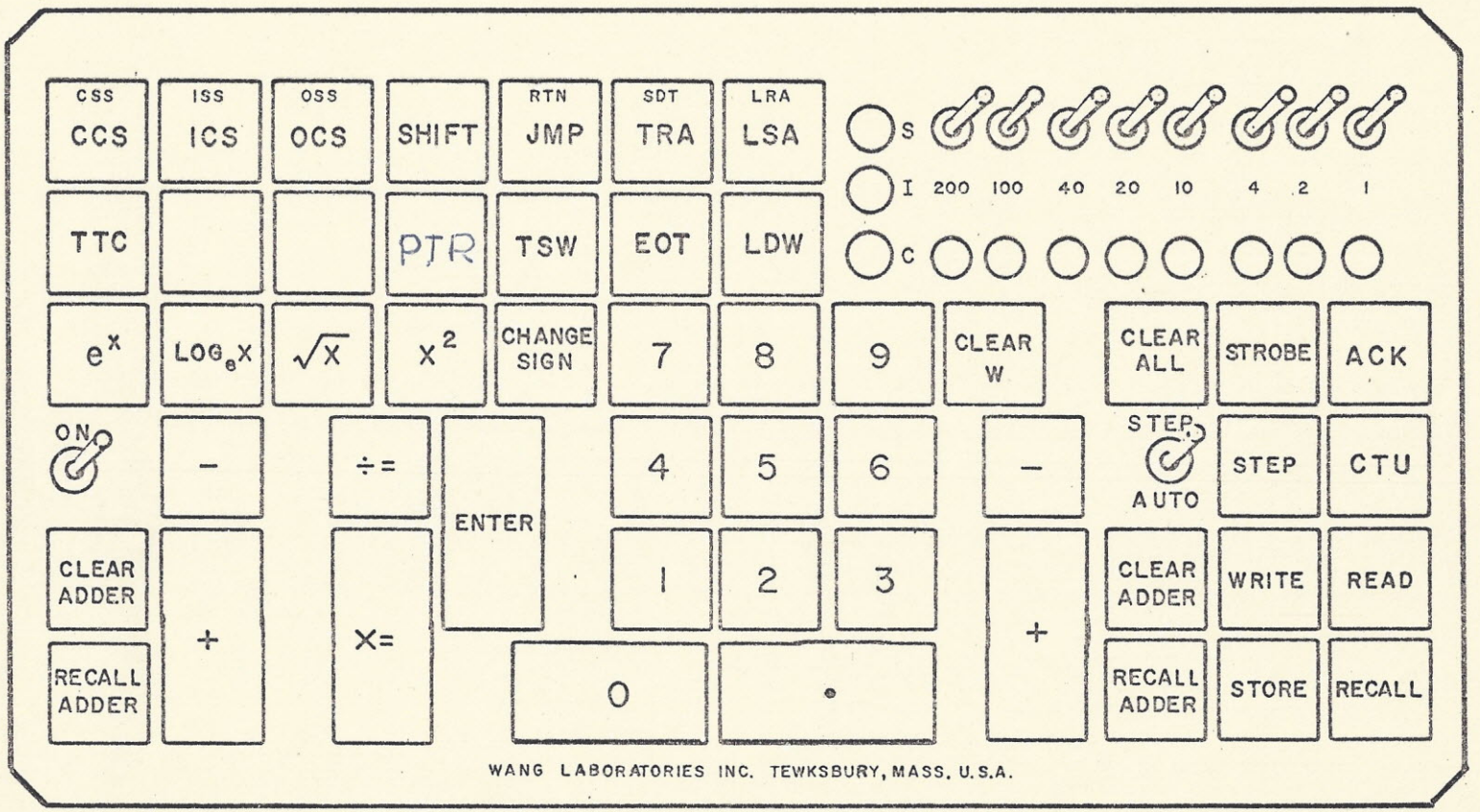
The teletype keyboard serves several purposes. Its keys may be used to provide typewritten information, to punch paper tape and to send data to the computer. The keys also control the operation of the teletype unit itself.

The CTRL key is used to initiate control operations. Its use is similar to that of the standard SHIFT key. Certain codes involve use of both the CTRL and SHIFT keys. For instance, punching is enabled using the CTRL and TAPE keys or disabled using CTRL and ~~TAPE~~. Printing may be suppressed using CTRL-Y and reactivated using CTRL-X. Teletype code tables provided in the appendix describe all necessary operations.

7.3 System Keyboard Controls

The keyboard console diagram appears in Figure 7.3. The keyboard unit is described in detail in Section 2.2

The ON switch controls power to the keyboard only. It is normally left in



KEYBOARD – 4000 SYSTEM FIG. 7.3

the up position.

The STEP-AUTO toggle switch controls program operation. In STEP the computer will execute a single instruction each time the associated STEP pushbutton is pressed. In this mode, the code for the command just executed can be read on the 8 bit lamp display. In AUTO the program will run automatically.

CLEAR ALL automatically resets all input, output, and control assignments, and establishes the keyboard as system control element. The CLEAR ALL key should not be used while a program is running in AUTO, as it may cause program information to be lost. The CLEAR ALL key clears all arithmetic registers but does not alter the contents of memory.

The eight code switches are used for testing standard codes and to generate busline codes for which no specific key exists. After the code pattern is set into the switch bank, the STROBE key is used to execute the operation. A switch in the up position specifies the presence of a code bit.

Notice that certain keys contain two code designations. The lower designation, in black, is the normal operation provided by the key. To obtain the upper operation, specified in red, hold down the SHIFT key, then press the appropriate key.

The ACK key generates a busline ACKNOWLEDGE pulse.

Lamps are provided to show the status of the buslines. Lamps also show when the KB is input (I) or control (C). Lamp S shows that the "STROBE" signal is on.

7.4 Loading Program Through The Keyboard

New programs are generally loaded into memory through the keyboard. Once the program has been tested, a tape is punched from memory to preserve the program for later reentry, when needed.

The procedure for loading the program is to assign the keyboard as input, the memory as output, and to specify where the first instruction is to be. A transfer (TRA) command will then establish a direct link from the keyboard to the memory, and the program can be keyed in. The final instruction in the program should be EOT, which will terminate the loading procedure. This addressing sequence is keyed in from the keyboard as follows:

```
CLEAR ALL
ICS
0        }        SELECT KB AS INPUT

OCS
7        }        SELECT MEMORY AS OUTPUT

OSS
0        }        SELECT LOCATION OF FIRST
                      INSTRUCTION

TRA
```

Once this is accomplished, notice that the keyboard "I" (input) lamp is on and that the memory control OUTPUT lamp is on. Notice also that the MEMORY ADDRESS display is set to zero. To load program, the operator simply presses the appropriate keys according to the program sheet.

As the program is keyed in, the MEMORY ADDRESS display will always show the address of the instruction just entered. Therefore, after instruction zero is keyed, the display still reads zero, and after instructions 1, 2, 3, 4 have

been keyed, the display will read 4. It is advisable to check this display from time to time, particularly when entering a long program, to guard against errors in the loading process. A convenient way of doing this is to check the display after entering the last instruction in each column of the program sheet, at which time the count should always end in 7 and the three rightmost lamps should be on.

If a mistake is detected, simply press EOT to end the transfer process, then address the point at which the correction is to be made (or the beginning of the column in which the mistake occurred) by keying

ICS 7

ISS
X
X
X
TRA

ADDRESS WHERE PROGRAM ENTRY IS TO RESUME

then continue entering the program from this point.

If the program being keyed into memory contains EOT codes, as would be the case if alphanumeric messages were stored as portions of the program, simply press TRA to continue loading after each EOT is entered. This is necessary since EOT automatically terminates the loading process.

Alphabetical information, or commands for which there are no keys, may be entered using the code switches with the associated STROBE key or by using the teletype keyboard after addressing the teletype unit as input device.

7.4.1 Instruction Modification

To change a single instruction, anywhere in memory, press:

CLEAR ALL

ICS
0 } KB IS INPUT

OCS
7 } MEMORY IS OUTPUT

OSS
X
X
X
X } ADDRESS OF INSTRUCTION TO BE MODIFIED (OR FIRST INSTRUCTION OF A SEQUENTIAL LIST TO BE MODIFIED)

TRA

then enter the new instruction (or the new list of sequential instructions), then press CLEAR ALL. Use of the clear all command ends the transfer operation without affecting the instruction which follows the one just modified. If EOT is used, it will replace that next instruction, and disrupt the program.

7.5 Program Operation

Once the program is loaded into memory, it may be started by assigning control to the memory unit, specifying where the program is to begin, and pressing the CTU key. The commands to accomplish this, which are issued from the keyboard are:

CLEAR ALL

CCS
7 } SET MEMORY AS CONTROL

CSS
X
X } LOCATION OF FIRST INSTRUCTION (1 TO 4 DIGITS)

CTU

If the STEP-AUTO switch is on AUTO, the program will run automatically.

In STEP, the program will execute one instruction each time the STEP key is pressed.

To stop a program, move the STEP-AUTO switch to STEP. Pressing CLEAR ALL will return control to the keyboard. To restart the program, use the addressing described above if CLEAR ALL has been used, or simply set STEP-AUTO to AUTO, if the memory is still in control.

7.6 Debugging

The STEP mode is useful in troubleshooting new programs. Using the step mode, it is possible to walk through the program, seeing the results at each step on the keyboard display.

In addition to the W register display, the operator can see the code of the instruction just completed. This code appears on the keyboard lamp display as well as on the OUTPUT BUFFER display of the Memory Control chassis. Also, the MEMORY ADDRESS display will show the address of the next instruction to be performed, and the MEMORY BUFFER display will show what that next instruction is.

When stepping through a program, the operator will notice what appear to be blank instruction codes occurring after each address sequence. This is a normal function of the address operation. When such blanks appear, simply press the STEP key to proceed.

To change an instruction, or list of instructions, is quite simple. Refer to Section 7.4 for details. However, adding instructions within a program is cumbersome because all of the instructions following those added must be moved.

One way of overcoming this is to remove some instructions from the area to be modified, then add a JMP sequence to branch the program to a modification routine. The modification routine then returns or jumps back to the main program. It is also possible to punch out the faulty program, make modifications by editing and reproducing a new tape on the teletype machine, then re-enter the program from tape.

7.7 Preparing Program Tapes

After a program has been keyed into memory, debugged and tested, a tape should be punched to preserve the program. The readout routine for punching this tape is in paragraph 7.7.2.

7.7.1 Preparation of Headings

It is often desirable to preface the program tape with headings, titles or operator instructions. This information will generally not be wanted in memory. Before dumping the program from memory into paper tape, this preface should be prepared manually from the teletype keyboard. Use the following procedure:

7.7.1.1 Turn LINE-OFF-LOCAL to LINE.

7.7.1.2 Activate punch with CTRL, TAPE keys.

7.7.1.3 Prepare a blank tape leader using the HERE IS key.

7.7.1.4 Enter the ~~STROBE~~ code by holding down CTRL and SHIFT and pressing N.

7.7.1.5 Enter the PRINT code by holding down CTRL and pressing X.

7.7.1.6 Type the headings and titles as desired.

7.7.1.7 Enter the ~~PRINT~~ code by holding down CTRL and pressing Y.

7.7.1.8 Enter the STROBE code by holding down CTRL and SHIFT and pressing O.

7.7.1.9 Enter a blank section by pressing HERE IS.

When the program tape is read by the computer, the "~~STROBE~~" code prevents the preface information from being sent to memory. Then after the headings are typed, the STROBE command allows the remainder of the tape to be read. The "~~PRINT~~" code prevents the program portion of the tape from being typed out.

7.7.2 Punching Tape From Memory

After the preface portion of the tape has been prepared manually, the program is transferred out with the following keyboard commands:

ICS 7	MEMORY IS INPUT
ISS X X X	LOCATION OF FIRST INSTRUCTION
OCS 6	TELETYPE IS OUTPUT
TRA	

The last code transferred will be EOT. If EOT appears more than once in the memory, the dumping process will stop after each. After each EOT, enter the X-OFF code on tape from the teletype keyboard, by holding down CTRL and pressing X-OFF.

Then press "HERE IS" to provide a blank portion of tape, and resume the program output by simply pressing TRA on the system keyboard.

After the final EOT is transferred to tape, enter the PRINT code, by holding CTRL while pressing X, before entering X-OFF as above. This activates the typewriter for possible program output. A final HERE IS will complete the program tape.

7.8 Loading Program From Tape

To read a program tape into memory, the following procedure is used:

7.8.1 Teletype LINE-OFF-LOCAL to LINE.

7.8.2 Load tape into tape reader by releasing plastic cover, placing tape so that feed holes engage sprocket teeth, closing cover. Tape advances toward front of reader.

7.8.3 Set reader START-STOP-FREE switch to STOP.

7.8.4 From system keyboard, press

ICS 6	TELETYPE IS INPUT
OCS 7	MEMORY IS OUTPUT
OSS X X	LOCATION OF FIRST INSTRUCTION (1 TO 4 DIGITS)
TRA	

7.8.5 If tape stops before end, press TRA again. Repeat if necessary until entire tape is read into memory. After the program is loaded, it may be started as described in Section 7.5.

LOCATION OF STORAGE REGISTERS IN MEMORY

MEMORY ADD.	STORAGE ADD.	MEMORY ADD.	STORAGE ADD.	MEMORY ADD.	STORAGE ADD.	MEMORY ADD.	STORAGE ADD.
0000	177	0400	137	1000	77	1400	37
0010	176	0410	136	1010	76	1410	36
0020	175	0420	135	1020	75	1420	35
0030	174	0430	134	1030	74	1430	34
0040	173	0440	133	1040	73	1440	33
0050	172	0450	132	1050	72	1450	32
0060	171	0460	131	1060	71	1460	31
0070	170	0470	130	1070	70	1470	30
0100	167	0500	127	1100	67	1500	27
0110	166	0510	126	1110	66	1510	26
0120	165	0520	125	1120	65	1520	25
0130	164	0530	124	1130	64	1530	24
0140	163	0540	123	1140	63	1540	23
0150	162	0550	122	1150	62	1550	22
0160	161	0560	121	1160	61	1560	21
0170	160	0570	120	1170	60	1570	20
0200	157	0600	117	1200	57	1600	17
0210	156	0610	116	1210	56	1610	16
0220	155	0620	115	1220	55	1620	15
0230	154	0630	114	1230	54	1630	14
0240	153	0640	113	1240	53	1640	13
0250	152	0650	112	1250	52	1650	12
0260	151	0660	111	1260	51	1660	11
0270	150	0670	110	1270	50	1670	10
0300	147	0700	107	1300	47	1700	7
0310	146	0710	106	1310	46	1710	6
0320	145	0720	105	1320	45	1720	5
0330	144	0730	104	1330	44	1730	4
0340	143	0740	103	1340	43	1740	3
0350	142	0750	102	1350	42	1750	2
0360	141	0760	101	1360	41	1760	1
0370	140	0770	100	1370	40	1770	0

SERIES 4000 CODE LIST
CODES 00-37
TELETYPE OPERATIONS

OCTAL CODE	PROGRAM NOTATION	OPERATION	TELETYPE CHARACTER
000			
001			
002			
003			
004	EOT	END OF TRANSFER	C EOT
005			
006			
007			
010			
011			
012	LF	LINE FEED	LINE FEED
013			
014	FORM	FORM FEED	C FORM
015	CR	CARRIAGE RETURN	RETURN
016			
017			
020			
021	X-ON	START READER	C X-ON
022	TAPE	ENABLE PUNCH	C TAPE
023	X-OFF	STOP READER	C X-OFF
024	-TAPE-	DISABLE PUNCH	C -TAPE-
025			
026			
027			
030	PRINT	ENABLE TYPEWRITER	C X
031	-PRINT-	DISABLE TYPEWRITER	C Y
032	SSR	SINGLE STEP READER	C Z
033			
034			
035			
036	-STROBE-	DISABLE STROBE	CS N
037	STROBE	ENABLE STROBE	CS O

C : USE CTRL KEY
S : USE SHIFT KEY

SERIES 4000 CODE LIST
CODES 100-137
SYSTEM OPERATIONS

OCTAL CODE	PROGRAM NOTATION	OPERATION	TELETYPE CHARACTER
100			
101			
102			
103	CTU	CONTINUE	C
104	TTC	TEMPORARY TRANSFER OF CONTROL	D
105			
106			
107			
110	CCS	CONTROL CHANNEL SELECT	H
111	CSS	CONTROL SUB-CHANNEL SELECT	I
112	ICS	INPUT CHANNEL SELECT	J
113	ISS	INPUT SUB-CHANNEL SELECT	K
114	OCS	OUTPUT CHANNEL SELECT	L
115	OSS	OUTPUT SUB-CHANNEL SELECT	M
116	RTN	RETURN	N
117	JMP	JUMP	O
120	SDT	SINGLE DIGIT TRANSFER	P
121	TRA	TRANSFER	Q
122	LDW	LOAD W	R
123	WRT	WRITE	S
124	RD	READ	T
125	TSW	TEST SIGN OF W (SKIP ON +)	U
126			
127			
130	STR	STORE	X
131	RCL	RECALL	Y
132	LSA	LOAD STORAGE ADDRESS	Z
133	LRA	LOAD RECALL ADDRESS	S K
134			
135			
136			
137			

S : USE SHIFT KEY

4K UNIT

CAPACITY FOR 32 PAGES

EACH PAGE CONTAINS 128 STEPS

EACH STORAGE REGISTERS TAKES 8 STEPS.

4096