

TRAINING FOR THE NO. 1 ESS

Matthew Raspanti
Bell Telephone Laboratories, Inc.
Holmdel, New Jersey

INTRODUCTION

The first installation of the No. 1 Electronic Switching System¹ (No. 1 ESS), a stored-program electronic telephone switching system developed by Bell Telephone Laboratories, started commercial operation in Succasunna, New Jersey, on May 30, 1965. In the No. 1 ESS, call processing and maintenance functions are performed under the control of a program of some 100,000 words stored in a read-only type of memory. The 44-bit instructions are executed by a central processor operating on a basic cycle time of 5.5 microseconds. A read-write type of memory with 24-bit word locations is used for the storage of transient information such as the digits dialed by a subscriber.

The introduction of the No. 1 ESS has made it necessary to train Bell System personnel in an entirely new technology. To meet this need, the author was charged with the responsibility of organizing an 8-month school on the No. 1 ESS to train a first group of 64 Bell System employees who would be involved in the installation or maintenance of the earlier central offices or in the training of personnel for subsequent installations. The school prerequisites were a high school education and a knowledge of basic electricity and electronics

(typically provided by operating companies in a 4-week course).

In the early stages of planning for the school, considering the complexity of the No. 1 ESS and the electromechanical background of the class, it was recognized that significant difficulties would be experienced by the trainees in understanding the basic concepts of program control and in visualizing the overall operation of the system, particularly its information-processing aspects. Also, since the school and the debugging of the system would be conducted concurrently, the 64 trainees could be expected to have only limited access to the No. 1 ESS for training purposes. In any case, it was felt that, because of its size, speed and complexity, the system itself would not be a suitable aid in the initial stages of training. The design of WOSP a Word-Organized Stored-Program training aid, was then undertaken to achieve the following objectives:

1. To develop a machine whose organization would conceptually parallel that of No. 1 ESS.
2. To provide full display of internal operation so that the information being processed, the operations being performed and

the flow of information would be directly apparent to the trainees.

WOSP is a miniature information-processing machine capable of completely automatic operation under the control of a program stored in punched cards such as the one shown in Fig. 1. Up to 4 cards can be used to provide 44 word locations of

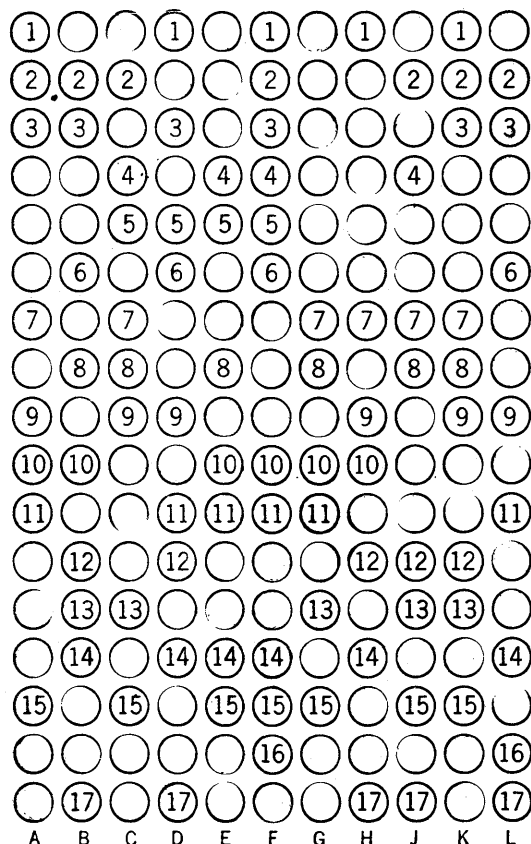


Figure 1. Program card.

17 bits each. Physically, WOSP consist of two units connected by a cable: a T-cart (Fig. 2) and a display panel (Fig. 3). The T-cart contains all the memory and logic circuitry which consists mainly of 4 card readers, about 170 wire spring relays, 48 reed relays and about 1,000 diodes. The four card readers and all the manual controls are mounted on the top of the T-cart. Relays and other electromechanical devices have been used because of their low cost and their readily understandable nature.

Five WOSP machines have been built so far, one by Bell Telephone Laboratories and four by Western Electric Company. These machines have been

used very successfully to introduce program control to several hundred trainees. Because of the opportunity it offers for truly "hands on" training, WOSP could be a valuable aid for programming training of a more general-purpose nature.

ORGANIZATION OF THE NO. 1 ESS SCHOOL

The No. 1 ESS school was set up as a special school to meet Bell System needs associated with the introduction of the No. 1 ESS. It was not intended to be a prototype for later schools to be conducted by the various telephone companies.

The 35-week trainnig, which started in January 1964, was conducted by a staff of 14 men, most of whom were also responsible for the preparation of maintenance manuals. Of the 14 men, 7 were Laboratories employees, 7 were on loan from Western Electric Company and 4 telephone companies. The 64 trainees came from 10 Bell System companies. The average age was 32, the average length of service was 10 years.

A typical day at the school included a 3-hour lecture delivered to all 64 trainees. During the remainder of the day, the trainees were divided into four groups, each permanently assigned to a study room under the supervision of a member of the staff. On the average, two hours were devoted to studying, one hour to class discussions and one hour to practice sessions in the school laboratory.

The general outline of the 35-week course is described below.

I—Fundamentals of Logic and Switching, 6 days

(Boolean notation, solid state logic, binary arithmetic, functional blocks such as registers, counters, scanners, selectors, etc., memory systems, stored logic)

II—Functional Description of No. 1 ESS, 4 weeks

(Principles of operation of the various system units, overall organization of the system program, handling of various types of telephone calls, maintenance facilities) This part was paralleled by a series of sessions using WOSP, as described later.

III—Circuit Fundamentals, 2 weeks

(Transistor circuits, pulse techniques, use of oscilloscopes, symbols and conventions of circuit drawings)

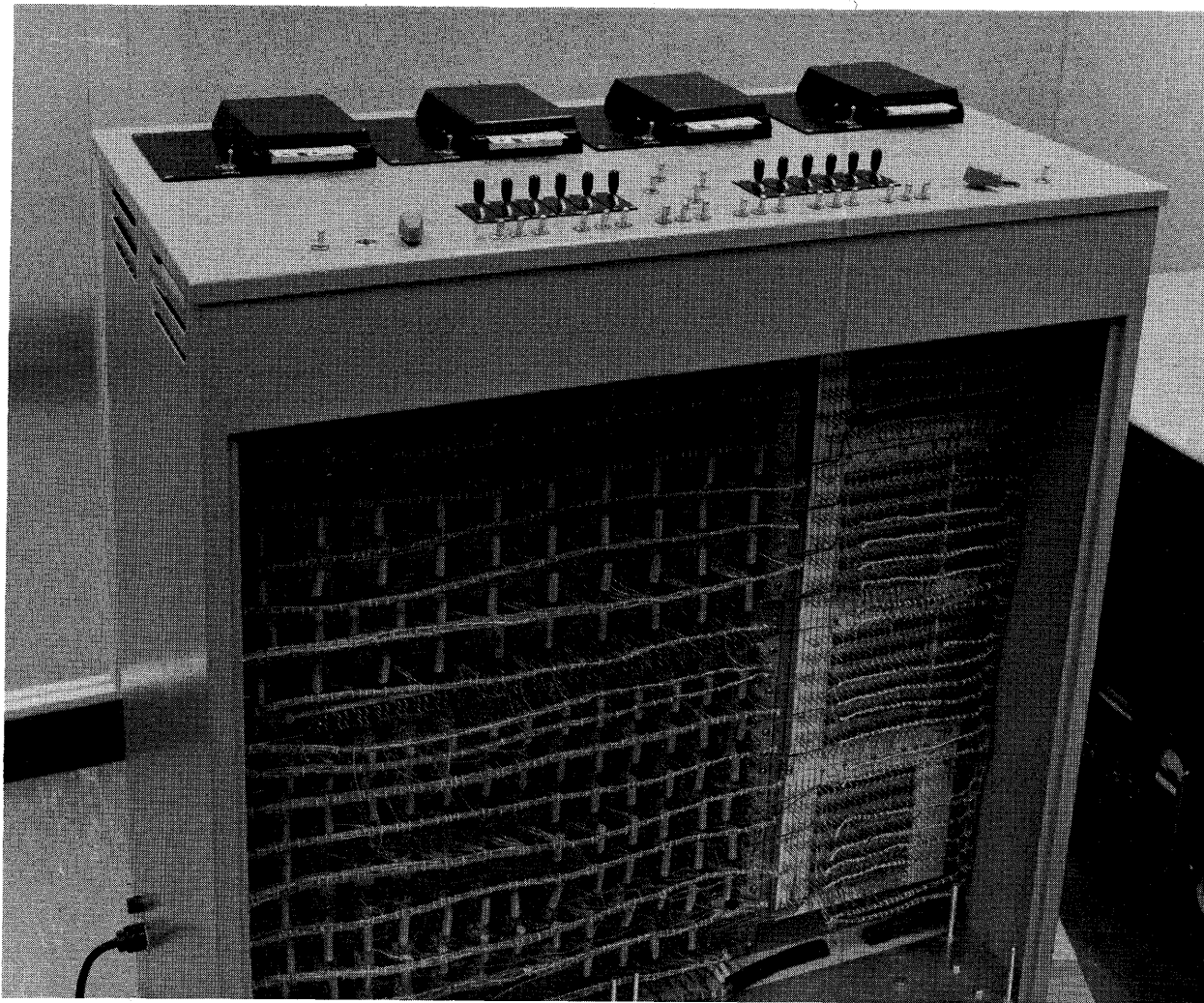


Figure 2. T-cart.

IV—No. 1 ESS Programming, *6 days*

(Detailed description of No. 1 ESS instructions and their use in typical programming situations)

V—Detailed Description of No. 1 ESS Circuits, *15½ weeks*

At the completion of each type of system unit, the associated installation test program was described at a functional level. A total of 6 days was devoted to these programs.

VI—Function Description of Executive and Call Processing Programs, *5 Weeks*

VII—Functional Description of Maintenance Programs—Administrative and Maintenance Procedures, *4 Weeks*

During the last months, in teams of 4 men, the trainees spent about 120 hours each in circuit laboratory sessions. In addition to four oscilloscopes, the equipment used in these sessions included several laboratory setups that had been excerpted directly from various major system units to provide the trainees with a representative cross section of devices and printed-wiring boards. In addition, toward the end of the school, the trainees were given a 2-week field assignment at a number of No. 1 ESS installation sites.

Starting in October 1964, a second school for 69 trainees was conducted for 32 weeks. The format followed was essentially similar to the one described for the first school.

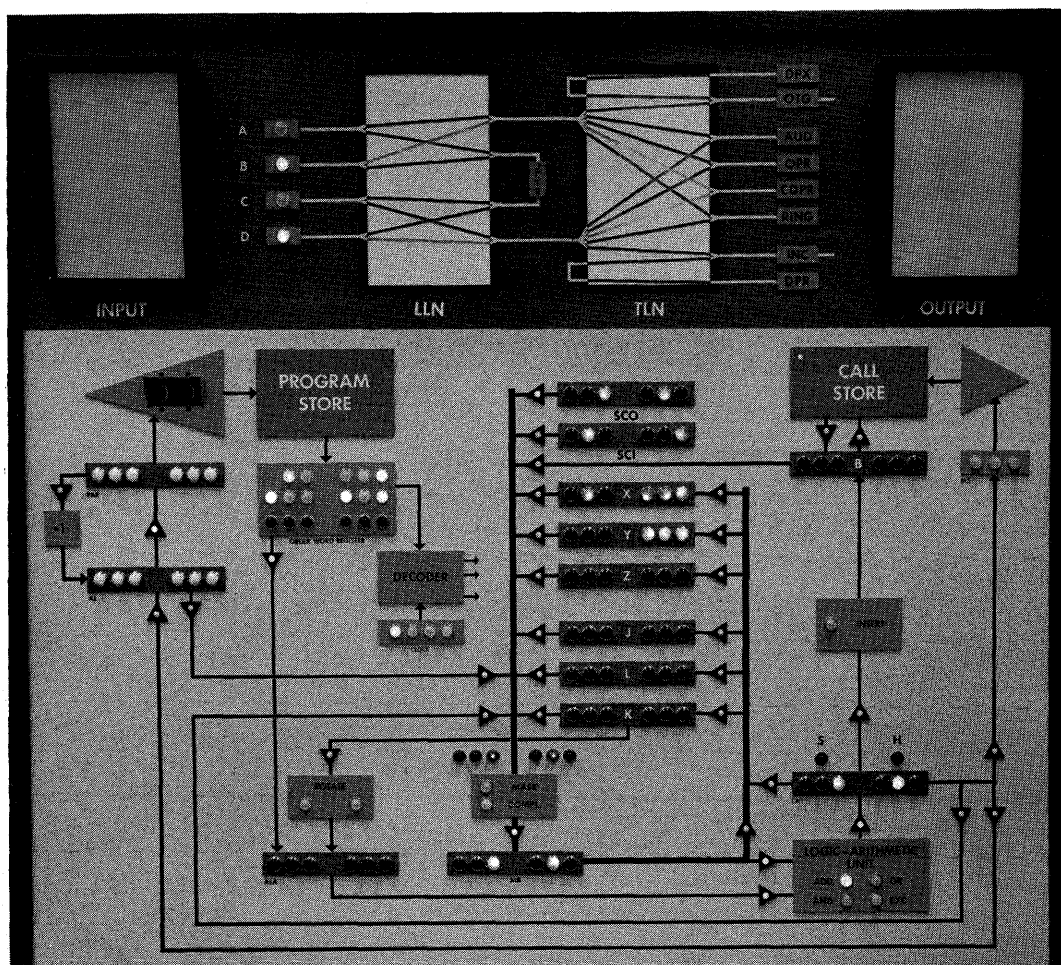


Figure 3. Display panel (side supports on casters not shown).

Classroom Use of WOSP

Two basically different approaches were adopted in the No. 1 ESS school with regard to hardware and software. The hardware was described in considerable detail to provide the necessary background for trouble-shooting. The software, on the other hand, was described mostly at a functional level except for a number of typical situations which were covered in complete detail. It was deemed necessary, however, to provide a good knowledge of program instructions and their use in order to develop, not a proficiency in programming, but a sound understanding of system operation and an ability to delve into program details as circumstances might require.

The most basic concepts of program control were introduced at the very end of Part I (Fundamentals of Logic and Switching). This was done with analo-

gies and with a simple paper machine having only four basic instructions. Use was then made of WOSP in a series of 14 one-hour class discussion sessions, which were repeated for each of two groups of 32 trainees. A carefully selected sequence of sample instructions and programs of increasing complexity was presented on the machine in a step-by-step fashion, repeating backtracking as necessary. Each trainee had a copy of the programs on which he could readily follow the operation of the machine since the address of the instruction under execution is displayed in octal form. In this fashion, many concepts, features, and terms basic to No. 1 ESS were gradually presented in a manner that had immediate significance to the class. At the end of the demonstration sessions, the trainees were given an opportunity to write some programs and to debug them on the machine. Practically all trainees individually wrote and debugged two of the three

programs that were suggested for this purpose. Several successfully undertook the writing of the third program which entailed storing in memory eight numbers and then rearranging them in ascending order.

During the WOSP sessions and later during part IV (No. 1 ESS Programming), a symbolic notation of the type described in the appendix was extensively used to describe the execution of program instructions. This approach proved very successful, particularly in the second school, even though the trainees did not have an extensive mathematical background. Most trainees found symbolic expressions more convenient than the often lengthy and awkward verbal descriptions of the conventional programmer's manuals, which were also made available to them.

WOSP has been used so far in the two sessions of the No. 1 ESS school and in several sessions of a course organized by Western Electric Company for its installation people. The experience gained definitely indicates that, with the aid of WOSP, trainees with no prior background can acquire in less time a more meaningful understanding of programming than with conventional methods.

WOSP HARDWARE

Functionally, WOSP consists of five major parts:

1. the Program Store—a random-access read-only memory for storing the program,
2. the Call Store—a random-access read-write memory for storing, updating, and retrieving data or control information,
3. the Processor which executes, one at a time, the instructions in the Program Store,
4. the input facilities,
5. the output facilities.

Program Store

The Program Store consists of 4 motor-driven Hickok Cardmatic card readers combined with a 44×17 diode matrix to provide random access to any one of 44 word locations of 17 bits each. Each card reader contains an 11×17 array of switches, one for each hole position on a card of the type shown in Fig. 1. When a card is inserted into a card reader, it causes the switches to remain open

where holes have been punched and to be closed where no holes have been punched.

The program memory described, beside being conceptually analogous to the program memory of the No. 1 ESS, offers the advantage of being easily and cheaply alterable in the classroom.

Call Store

The Call Store consists of an 8×6 array of reed relays with controls to provide random access to 8 memory locations of 6 bits each for either reading or writing.

Processor

The Processor, whose internal block diagram is fully depicted on the display panel (Fig. 3), contains:

- (a) facilities for communicating with the program store and the call store,
- (b) facilities for decoding and executing individual program instructions, and for determining the next program address,
- (c) a number of 6-bit registers for detecting inputs, for controlling outputs or for general-purpose storage of information,
- (d) facilities for performing arithmetic additions and logical operations (logical product, logical union, exclusive OR, complementation, and circular shift). The 6-bit words to be operated on may be stored in registers or in call store locations.

Regardless of type, the execution of an instruction requires a single machine cycle consisting of four clock phases. Five different modes of operation are available:

- (a) Instructions are executed at the rate of two per second.
- (b) Instructions are executed at the rate of one per second.
- (c) Each instruction is executed during a 1-second interval. Upon completing the execution, the machine stops and waits for a command from the operator to proceed. When given this command through a cord switch, WOSP proceeds to execute the next instruction, upon completion of which it stops as before.

- (d) Instructions are executed with the clock under the control of the cord switch. Thus, the duration of each of the four clock phases is under the direct control of the instructor or trainee operating the machine.
- (e) In this mode, 17 toggle switches on the T-cart top panel are used for the manual setting of the instruction to be executed. The control of the clock is as in (d) above.

At any time during the execution of a program, the machine can be stopped at the completion of the instruction in progress; if desired, a different mode of operation may be selected before operation is resumed. Thus, for tutorial or testing purposes, the operation of the machine may be followed at different levels of detail.

Output Facilities

The OUTPUT section in the upper right corner of the display panel is a 6×2 array of windows and lamps. Through a slot, a transparent sheet with appropriate legends can be placed in front of the windows so that, when the lamp behind a particular window is lighted, a message is presented by WOSP. By means of appropriate display instructions, the programmer can request that certain OUTPUT windows be placed under the control of register X, Y or Z. The 6 windows on the left can be controlled by the 6 bits of register X; of the 6 windows on the right, 3 are controlled by the right half of register Y, 3 by the left half of register Z.

In the upper center portion of the display panel, the two units designated LLN (Line Link Network) and TLN (Trunk Link Network) make up the network display. This display is used as an output for programs processing simulated telephone calls through an elementary switching network. As a "call" progresses, approximate paths are displayed through the line link network and trunk link network. By means of appropriate network instructions, the programmer can request that certain network paths be illuminated under the control of register X, Y, or Z. Figure 4 shows the association between the network paths and the bits of registers X, Y and Z.

Input Facilities

In one mode of operation, WOSP has 12 inputs or scan points that can be independently controlled

by means of 12 keys on the top panel of the T-cart. These keys are labeled SC00 through SC05 and SC10 through SC15. All inputs are detected internally through two 6-bit scanner registers, SC0 and SC1. The bits of registers SC0 and SC1 can be displayed by lamps in the 12 windows of the INPUT unit in the upper left of the display panel. A transparent sheet can be placed in front of the windows to indicate by means of appropriate legends the significance of the inputs for the particular program that is controlling the machine.

In another mode of operation, WOSP has only six scan points that can be independently controlled by means of the six keys SC00 through SC05. Of these, keys SC00 through SC03 control the state of the four telephone "lines" A, B, C, and D connected to the network (Fig. 3). There are six more scan points that are controlled by keys SC00 through SC03 via paths established through the network. The six scan points are associated with the six network terminals to the circuits labeled JCTR (Junction Circuit), AUD (Audible Tone Circuit), OPR (Operator Trunk), CDPR (Customer Dial Pulse Receiver) and RNG (Ringing Circuit). For example, at the scan point associated with the CDPR, it is possible to detect the state of whichever of the four lines A, B, C, or D is connected to the CDPR via the network. Figure 4 shows the association between the bits of registers SC0 and SC1 and the scan points provided for the network terminals.

The Program Store, the Call Store, the Processor, the Line Link Network and the Trunk Link Network have direct conceptual counterparts in the No. 1 ESS. Although speed, capacity and logical complexity are vastly different, WOSP is fairly representative of the overall organization and operation of No. 1 ESS, particularly within the Processor.

BASIC TYPES OF INSTRUCTIONS

The 17 bits that make up an instruction can be divided as follows:

- (a) The 11 most significant bits specify the operation to be performed, including certain variations that can be optionally requested by the programmer.
- (b) The remaining 6 bits, referred to as the data-address (DA) part of the instruction, usually specify data to be operated on, or the address of an instruction or of a data word.

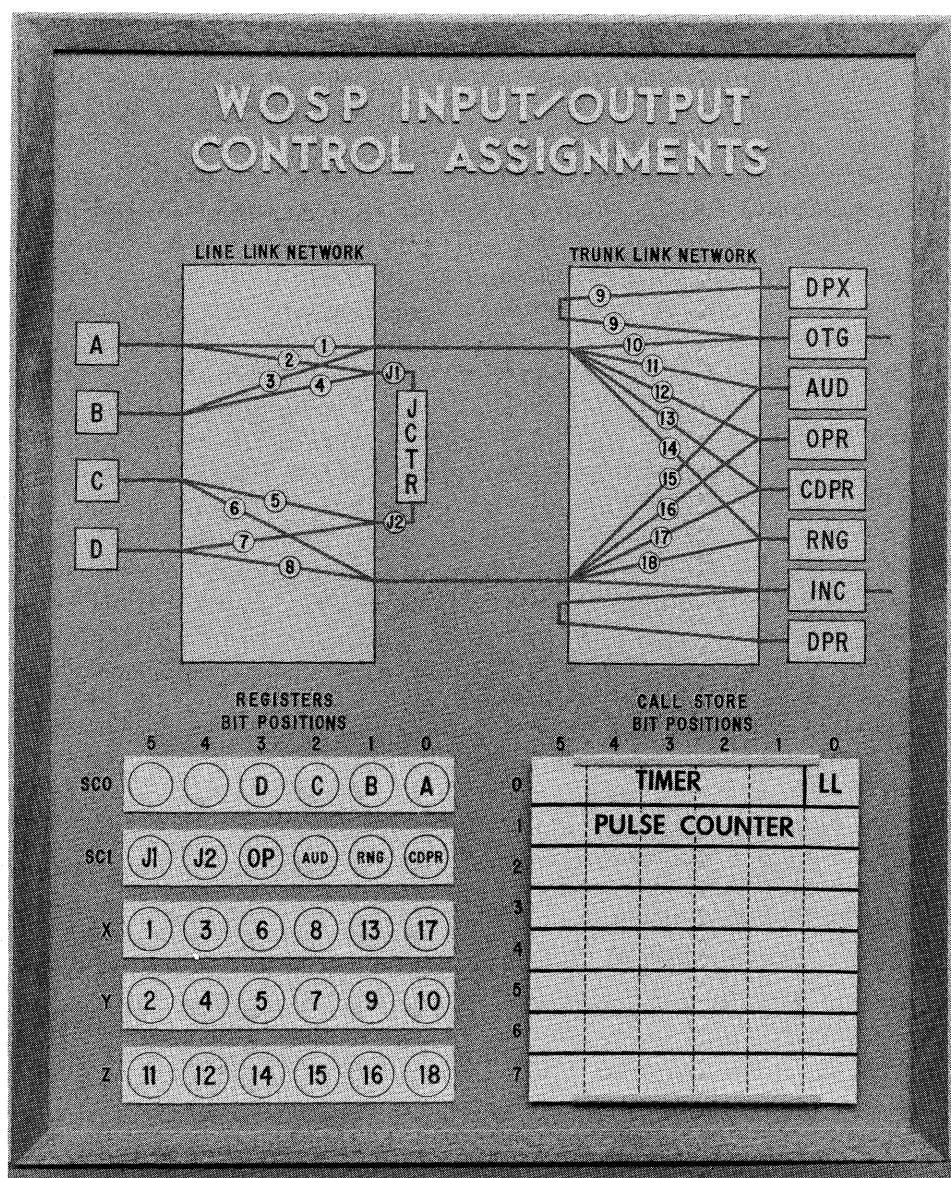


Figure 4. Input/output assignments.

An instruction in which the programmer has made use of the indexing option is executed using an effective DA field obtained by adding the contents of the DA field and those of a specified index register.

Let W represent the effective DA field. Also, let M represent the memory (call store) location specified by W. The types of instructions available in WOSP can then be broadly described as follows:

1. W-to-Register—Move W into a specified register.

- *2. W-to-Memory—Move W into a specified call store location.
3. Register-to-Memory—Move the contents of a specified register into M.
5. Add (Subtract)—
 - (a) Add (subtract) W to (from) the contents of accumulator register K and store the result in K.
 - (b) Same as 5(a) using the contents of M instead of W.
6. Logical Operations—
 - (a) Determine the logical product

- (AND) of W and the contents of register K and store the result in K.
- (b) Same as 6(a) using the contents of M instead of W.
- (c) Same as 6(a) and 6(b) for the operations of logical union (OR) and exclusive OR.
7. Shift—
- (a) Shift the contents of register K to the left (or right) by 1 or 2 positions.
- (b) Rotate the contents of register K to the left (or right) by 1 or 2 positions.
- *8. Output Control—Control the network display or the OUTPUT display.
9. Unconditional Transfer—Transfer to program address W. If specified, remove part or all of a network or OUTPUT display.
10. Conditional Transfers—
- (*a) Transfer to program address W if a specified bit of register K is 0.
- (b) Transfer to program address W if, for the result of some previously performed operation, some specified condition is satisfied by the sign S (= 1 if the result is negative) and/or the homogeneity H (= 1 if the bits of the result are all equal to 0 or all equal to 1).

Examples of transfer conditions in terms of S and/or H are:

P	Plus	S = 0
M	Minus	S = 1
AZ	Arithmetic zero (All zeros or all ones)	H = 1
AU	Arithmetic "unzero"	H = 0
LZ	Logical zero (all zeros)	S = 0 and H = 1
LU	Logical "unzero"	S = 1 or H = 0
LE	Less than or equal to arithmetic zero	S = 1 or H = 1
GE	Greater than or equal to arithmetic zero	S = 0 or H = 1

The instructions described above have direct counterparts in the instruction list of No. 1 ESS except for the items marked by an asterisk. (There

are several other types of instructions in No. 1 ESS of a general purpose nature or to control input, output, maintenance and other specialized functions.)

In addition to indexing, the programmer can avail himself of one or more of the following options.

C Option (Complementing). When this option is used, a data word is complemented before undergoing some arithmetic or logical operation.

PL Option (Product Masking). When this option is used, a data word is ANDed with the contents of register L. Some unwanted bits can thus be masked out before some logical or arithmetic operation is performed.

EL Option (Insertion). At times, it is necessary to insert information in some bit positions of a call store location without affecting in any way the remaining bit positions. For this reason, instructions for writing into the Call Store include as an option the insertion of a word A into a location B through a mask m. In each bit position where m has a 1, the bit of A replaces the bit initially in B. In each bit position where m has a 0, the bit initially in B remains unchanged. The mask m is provided by register L.

J Option. Whenever a transfer to some address X is specified at some address P, the programmer has the option of requesting that address P + 1 be saved in the J register. Thus, if the transfer is made to a subroutine, the use of the J option makes it possible to request at the end of the subroutine that the program return (jump back) to P + 1.

A complete description of the program instructions and their options is given in the Appendix in symbolic form.

ILLUSTRATIVE PROGRAMS

Call Processing Program

A 44-word program has been written to process "telephone calls" originated by any one of the four lines A, B, C or D (Fig. 4). Line A or B may call C or D and vice versa. The "directory numbers" of lines A, B, C and D are 1, 2, 3 and 4 respectively. Line changes associated with originations, dial pulses and disconnects are generated by manipulating keys SC00 through SC03 for lines A, B, C and D respectively. For instance, to simulate B calling D, key SC01 first is turned on to originate the call, then is turned off and on four times to "dial"

D's directory number and finally is turned off to disconnect. Only one call can be connected at any time.

The program performs the following functions:

1. By scanning periodically the four lines, identify which line, if any, has originated a call.
2. Connect the calling line through LLN and TLN to the customer dial pulse receiver (CDPR).
3. By scanning periodically the CDPR, detect and count the dial pulses generated by the calling line.
4. Recognize the completion of the single digit dialed.
5. Through the LLN and TLN, connect the calling line and the called line to AUD and RNG respectively. (In a real situation, AUD would return an audible tone to the calling line, RNG would apply ringing to the called line.)
6. By scanning RNG and AUD, detect an answer by the called line or an abandon by the calling line.
7. If the calling line abandons, release the network connections and start scanning again for an origination.
8. If the called line answers, release the network connections previously set up. Through the LLN establish a talking connection between the calling and called lines via the junctor.
9. By scanning both sides of the junctor, detect when either of the two lines disconnects. Release the network connections and start scanning again for an origination.

Two words in call store are used to process a call, see Fig. 4. Bit 0 of word 0 is used as a last look (LL) bit to record the previously observed state of the scan point for CDPR. Bits 1, 2 and 3 are used as a timing counter which is recycled whenever a dial change is detected at CDPR and is incremented whenever no dial change is detected at CDPR. Only after the completion of dialing will the counter be incremented until bit 3 becomes 1.

Call store word 1 is used as a pulse counter. Initially, bit 0 is made a 1. Whenever a dial change from 0 to 1 is detected, the contents of the pulse counter are rotated to the right by one position. At the completion of dialing, the pulse counter will

have a 1 in bit position 5, 4, 3, or 2 depending on whether the dialed digit was 1, 2, 3 or 4 respectively.

Figure 5 shows that part of the program that controls the detection of dial changes, the counting of dial pulses and the timing to detect dialing completion. The program reaches octal address 14 after an originating line has been identified and connected to the CDPR through the LLN and the TLN.

For each instruction in Fig. 5, the first line shows the octal address and the instruction itself in symbolic form. The second line describes symbolically the processing actions caused by the instruction. The following notation has been used here:

COMP(X)	= Complement of X
AND(X,Y)	= Logical product of X and Y
OR(X,Y)	= Logical union of X and Y
EXCL.OR(X,Y)	= Exclusive OR combination of X and Y
X→Y	= X is moved into Y
M1	= Call store location #1
K0	= Bit 0 of register K

The intent of each instruction is explained after its symbolic description.

Call Store Test Program

This program (Fig. 6) is representative of some of the techniques used in testing the No. 1 ESS. Its purpose is to test the ability to read and write properly at each of the eight memory locations of the call store. The operator can set up any two 6-bit words SC0 and SC1 on the 12 input keys of the T-cart. Test words SC0 and SC1 are alternately copied into the call store so that SC0 is written into locations 0, 2, 4, and 6 and SC1 is written into locations 1, 3, 5 and 7. Upon completion of this loading stage, the test proper starts. The contents of the call store locations are read out one at a time and compared with the appropriate test words. Whenever a discrepancy is found between the word from the call store and the associated test word, the machine stops and gives an OUTPUT display to indicate that a failure has been detected. Note that the bit(s) at fault can be precisely pinpointed since both the word read out and its address are shown on the display panel. The operator instructs the machine to resume the testing by pressing the cord switch. When the last call store location has been tested, the machine stops and gives an OUTPUT display to indicate that the test has been completed.

- | | |
|--|--|
| <p>(14) WML 000 001
000 001 → M1
Initialize pulse counter in M1 (Pulses will be counted by rotating the 1 bit to the right by one position for every pulse dialed)</p> <p>(15) WMO 000 001
000 001 → MO
Reset the timer and write 1 in Last Look (LL) bit in MO</p> <p>(16) WK 0, SC1
(0 + SC1) → K
Read scan point CDPR in SC1 and store in K</p> <p>(17) XMK 0
EXCL. OR(MO, K) → K
Excl. OR combination of Last Look bit in MO and scan pt. CDPR makes KO equal to 1 whenever there is a dial change</p> <p>(20) TKBO 31
Tra. to 31 if KO=0
Tra. occurs if no dial change has occurred</p> <p>(21) WMO 0
0 → MO
Reset the timer in MO since a dial change was detected. Reset the Last Look bit for a possible down change.</p> <p>(22) WJ 0, SC1
(0 + SC1) → J
Read scan pt. CDPR in SC1 and store in J</p> <p>(23) PWK 0, J
AND (0 + J, K) → K
AND combination of scan pt. CDPR and change bit in KO makes KO equal to 1 if dial change was upward</p> | <p>(24) TKBO 16
Tra. to 16 if KO=0
Tra. occurs if dial change was down (Last Look bit was changed to 0 at step 21)</p> <p>(25) MK 1
M1 → K
Read M1 into K to increment the pulse counter</p> <p>(26) QC
Rotate K to the right by one position</p> <p>(27) KM 1
K → M1
Store incremented pulse count back in M1</p> <p>(30) T 15
Tra. to 15</p> <p>(31) MK 0
MO → K
Since no dial change has occurred, read MO into K to increment the dialing completion timer</p> <p>(32) AWK 2
(2 + K) → K
The timer is incremented by adding 2 so as not to affect the Last Look bit</p> <p>(33) KM 0
K → MO
The incremented time count is stored in MO</p> <p>(34) TKB3 16
Tra. to 16 if K3=0
Tra. occurs as long as the time count is less than 4</p> |
|--|--|

Figure 5. Telephone call processing program.

ACKNOWLEDGMENTS

The author wishes to acknowledge the contributions of R. E. Osteen who designed the clock counter, the mode selection logic and the test program for debugging the machine.

REFERENCES

1. *Bell System Technical Journal*, special issue on No. 1 ESS, Sept. 1964.

APPENDIX

A SYMBOLIC DESCRIPTION OF WOSP ORDERS

The general format of WOSP orders is similar to that of No. 1 ESS orders. It consists of an operation field, which specifies the type of processing action to be performed, and a variable field, which is divided into three subfields: DA, R, and LCJ. The DA subfield specifies a number to be interpreted as data or as an address. The R subfield specifies the

(00) START	WX	111 000	Initialize index register X to -7 for loading.
(01)	UWDYS	000 011	Stop and display the message "TEST COMPLETED-SET KEYS."
(02)	TUD010	LOAD	Remove the display and transfer to address (03).
(03) LOAD	SCOM	7,X	Move test word from SCO to call store.
(04)	WX	1,X	Increment index register X.
(05)	SC1M	7,X	Move test word from SC1 to call store.
(06)	WX	1,X	Increment index register X.
(07)	TCM	LOAD	Check for completion of loading. Transfer to (03) if X is still negative.
(10)	WX	111 000	Initialize index register X to -7 for testing.
(11) TEST	WK	0, SCO	Move test word from SCO to K.
(12)	CMK	7,X	Compare test word with memory readout.
(13)	TCAU	ERROR, ,J	Check for failure. Transfer to (23) if memory readout \neq test word. Because of the J option, return address (14) is stored in register J.
(14)	WX	1,X	Increment index register X.
(15)	WK	0,SC1	Move test word from SC1 to K.
(16)	CMK	7,X	Compare test word and memory readout.
(17)	TCAU	ERROR, ,J	Check for failure. Transfer to (23) if memory readout \neq test word. Because of the J option, return address (20) is saved in register J.
(20)	WX	1,X	Increment index register X.
(21)	TCM	TEST	Check for completion of test. Transfer to (11) if X is still negative.
(22)	T	START	Return to START.
(23) ERROR	UWDYS	000 100	Stop. Display the message "FAILURE."
(24)	TUD010	0,J	Remove the message display and resume testing at either (14) or (20).

Figure 6. Call store test program.

register to be used when indexing is called for. The LCJ subfield is used to specify program options other than indexing: product masking (PL), insertion (EL), complement (C), and save return address (J).

A complete symbolic description of all WOSP orders is given in Fig. 7. Particularly for complex machine languages such as that of No. 1 ESS, the type of symbolic notation used in Fig. 7 is very convenient for a description of instruction execution that is at the same time very precise and concise.

Definitions

D	Contents of DA subfield
R	Contents of register specified by the R subfield
W	Effective DA subfield ($= D + R$)
M	Contents of call store memory location identified by W

X	Contents of register X
X'	Complement of binary word X
$X + Y$	Arithmetic sum of X and Y
$X - Y$	Arithmetic difference of X and Y
$X \cdot Y$	Logical product of X and Y
$X \perp Y$	Logical union of X and Y
$X \times Y$	Exclusive OR of X and Y
$X \nearrow N$	Contents of register X shifted to the left of N bit positions.
$X \searrow N$	Contents of register X shifted to the right of N bit positions.
$X \leftarrow N$	Contents of register X rotated to the left of N bit positions.
$X \rightarrow N$	Contents of X rotated to the right by N bit positions.
$X \rightarrow Y$	Word X replaces the contents of location Y. If X represents the contents of some register or memory location, it remains unchanged in its initial location.

INSTRUCTION FORMATS WITH OPTIONS		EXECUTION DESCRIPTION
NETWORK & DISPLAY	PWD \underline{X} D, R ₂	(Y, Z) IF i=1: D-R ₂ →X, CF / GIVE OUT-IF i=0: R ₂ →X, CF / PUT DISPLAY
	PWD \underline{X} S D, R ₂	(Y, Z) SAME AS PWD \underline{X} . ALSO, STOP
	PWN \underline{X} D, R ₂	(Y, Z) IF i=1: D-R ₂ →X, CF / GIVE NET-IF i=0: D→X, CF / WORK DISPLAY
	UWD \underline{X} D, R ₂	(Y, Z) IF i=1: DLR ₂ →X, CF / GIVE OUT-IF i=0: D→X, CF / PUT DISPLAY
	UWD \underline{X} S D, R ₂	(Y, Z) SAME AS UWD \underline{X} , ALSO STOP
	UWN \underline{X} D, R ₂	(Y, Z) IF i=1: DLR ₂ →X, CF / GIVE NET-IF i=0: D→X, CF / WORK DISPLAY
SHIFT ROTATE	H	(K≠1)→K, CF
	Q D	(K≠1)·D→K, CF
	H2	(K≠2)→K, CF
	Q2 D	(Q≠2)·D→K, CF
	HC	(K≠1)→K, CF
	QC D	(K≠1)·D→K, CF
TRANSFER	H2C	(K≠2)→K, CF
	Q2C D	(K≠2)·D→K, CF
	TCP D, R, J	(M, LZ, LU, AZ, AU, LE, GE) IF CF=P: J→W IF (CF=P)·(j=1): PAR+1→J
	TKB \underline{S} D, R, J	(0, 1, 2, 3, 4) IF K5=0: J→W IF (K5=0)·(j=1): PAR+1→J
	TUN $\underline{a}, \underline{b}, \underline{c}$ D, R, J	J→W; IF j=1: PAR+1→J REMOVE NETWORK DISPLAY. NOTE 1
	TUD $\underline{a}, \underline{b}, \underline{c}$ D, R, J	J→W; IF j=1: PAR+1→J REMOVE OUTPUT DISPLAY. NOTE 1
REGISTER	T D, R, J	J→W; IF j=1: PAR+1→J

NOTE 1.
 IF a=1 REMOVE DISPLAY CONTROLLED BY X
 IF b=1 REMOVE DISPLAY CONTROLLED BY Y
 IF c=1 REMOVE DISPLAY CONTROLLED BY Z

INSTRUCTION FORMATS WITH OPTIONS		EXECUTION DESCRIPTION
M O V E	W \underline{X} D, R ₁ , PLC	(J, K, L, Y, Z) [W·L ^p] ^c →X, CF
	WM \underline{Z} D	(0, 1, 2, 3, 4, 5, 6) W→B, M7, CF
	XM D, R, PLC	(SC0, SC1, J, K, L, Y, Z) IF e=0: [X·L ^p] ^c →B, M, CF IF e=1: X ^c ·L·B·L ^p →B, M; X ^c →CF
	M \underline{X} D, R, PLC	(J, K, L, Y, Z) M→B; [M·L ^p] ^c →X, CF
	MC D, R, PLC	M→B; [M·L ^p] ^c →CF
A D D	AWK D, R, PL	K±[W·L ^p] ^c →K, CF
	SWK	
	AMK D, R, PL	M→B; K±[M·L ^p] ^c →K, CF
	SMK	
	CWK D, R, PL	K-[W·L ^p] ^c →CF
	CMK D, R, PL	K-[M·L ^p] ^c →CF
L O G I C	PWK D, R, PLC	K·[W·L ^p] ^c →K, CF
	UWK D, R, PLC	K⊥[W·L ^p] ^c →K, CF
	XWK D, R, PLC	K⊙[W·L ^p] ^c →K, CF
	PMK D, R, PLC	M→B; K·[M·L ^p] ^c →K, CF
	UMK D, R, PLC	M→B; K⊥[M·L ^p] ^c →K, CF
	XMK D, R, PLC	M→B; K⊙[M·L ^p] ^c →K, CF
E E		NO OPERATION

R = SC0, X, J, K R₁ = SC0, SC1, X, Y, Z, J, K, L R₂ = X, Y, Z, K

Figure 7. Symbolic description of WOSP orders.

X→CF	The sign and homogeneity of word X determine the states of control (C) flip-flops S and H respectively.
W	Transfer to program store address W.
IF a = 1:B	If Boolean variable a is 1, then expression B applies.
i	Boolean variable equal to 1 if the indexing option is used, that is, if a register is specified in the R subfield.
p,e,c,j	Boolean variables which are equal to 1 when options PL, EL, C, and J respectively are used.
R·L ^p	If p = 1, R·L ^p = R·L If p = 0, R·L ^p = R
X ^c	If c = 1, X ^c = X' If c = 0, X ^c = X
CF = LZ	The states of the control flip-flops S and H indicate a logical zero (LZ) condition (S = 0, H = 1). Similarly "CF = P" means that a plus (P) condition exists (S = 0).

As an example, in Fig. 7 the entry

WX D, R₁, PLC (J, K, L, Y, Z)

serves to describe the set of similar orders WX, WJ, WK, WL, WY, and WZ. The execution description is given only for WX and is interpreted as follows:

The result of indexing, W, possibly masked and possibly complemented is used to set the X register and the S and H flip-flops.

The LCJ subfield of an entry in Fig. 7 names the available options (other than indexing) for the class of orders described by that entry. For example, PLC in the LCJ subfield indicates that only the options of product masking (PL) and complementing (C) are available.

The orders grouped under the category Network and Display are the orders that selectively place the network display (LLN and TLN) or the OUTPUT display under the control of the X, Y, and registers, after performing an optional AND or OR operation as described in the execution description. ANDing makes it possible to remove part of a display previously established; O Ring makes it possible to add to a display previously established.

As an example, consider

PWNX 010010,K.

Since the R subfield is not blank, i = 1. Following the execution description of Fig. 7, the logical product 010010:K = 0K400K,0 replaces the contents of the X register and sets the C flip-flops. Since X appears in the operation field, network paths 1, 3, 6, 8, 13 and 17 (see Fig. 4) come under the control of bits 5, 4, 3, 2, 1 and 0 respectively of the X register. In this example, network paths 1, 6, 8, and

17 will not be illuminated, network paths 3 and 13 will be if bits X_4 and X_1 respectively are equal to 1.

Once an area of the network display or the OUTPUT display is placed under the control of the X, Y, or Z register, it remains under that control until

removed therefrom by an order of the TUN or TUD type. An area of the network or OUTPUT display, when not under the control of its associated register, is all dark regardless of the state of the register.

