# M E M O R A N D U M

TO: TX-0 Computer Users

FROM: Earle W. Pughe, Jr.

SUBJECT: Preliminary Programming for TX-0

DATE: July 23, 1958

---

The M.I.T. Lincoln Laboratory has given the TX-0 computer to the M.I.T. Electrical Engineering Department on an extended loan basis. The computer is intended to be used as a research and educational tool. The TX-0 computer is now being installed in Room 26-248 and should be ready for use by September 15, 1958.

This memo is written as an aid to those who wish to write programs for the computer before the computer is in operation. Interested users are also refered to Lincoln Laboratory Memorandum 6M-4789 entitled, "A Functional Description of the TX-0 Computer" written by John T. Gilmore.

The TX-0 computer has 4096 words of magnetic core storage. The cycle time is six (6) microseconds, thus each order normally will take twelve (12) microseconds. The inputs consist of a direct typewriter and a photo-electric tape reader (PETR). The outputs are a typewriter, a paper tape punch and a display scope. Other inputs and outputs are the toggle switch register and indicator lights on the control console. Provision has been made for users to connect their own equipment to the computer.

As of this date, there is no checked-out program to read in Flexowriter tapes. There is, however, a direct input typewriter

routine written by Mr. Charles Woodward, that will produce a suitable binary punched paper tape. This binary tape can then be read into the computer without further reference to the typewriter . It is also expected that a Flexowriter-to-computer language program will be ready by Sept. 15. The conversion program is now being written under Prof. Dean Arden by Mr. Monroe Weinstein.

It is expected that in the future the TX-0 computer will have more orders and more memory. Every effort will be made not to obsolete existing programs as new features are added. However, to help meet the objective of not obsoleting programs as the computer is modified, the unused bits of an instruction must be zeros. This restriction means that such tricks as shifting a word to change instructions will obsolete a program when changes are made to the computer. Bits "0" and "1" are now used for the instruction, bits "6" thru "17" are now used for the address and bits "2", "3", "4" and "5" must be zero for all orders except "operate" (opr).

In the TX-0 computer room (26-248) there is a complete set of drawings and memoranda referring to the computer. These files may be used by all interested persons, but it is imperative that no one remove any of the information from the room.

The Ad Hoc Committee on Experimental Computation (Chairman: Prof. J. F. Reintjes) is the faculty group in charge of the computer and they have final decision as to who may use the computer. It is expected that with the cooperation of the users there will be a minimum of paper work in assigning computer time. Since the computer is to be used for experiments instead of for numerical computations, the blocks of assigned computer time will be considerably longer than with other types of computers.

Appendix No. 1 describes the instructions used in the normal mode of operation, including the operate class instructions, and has a chart showing the address in octal of each point on the display scope. Appendix No. II describes the proposed conversion program which allows a programmer to use symbolic addresses in a program prepared on a Flexowriter tape. Appendix No. III describes the utility system which allows the user to have direct communication with the program in the computer.

It is expected that future memoranda will be issued to supplement the information contained in this memorandum, so it is suggested that the users keep this memorandum.

Earle W. Pughe

Earle W. Pughe

cc: Ad Hoc Committee on
Experimental Computation

Professors Brown, Shannon, Fano,
Rosenblits, Reintjes,
Fletcher, Susskind and
Mr. Wesley Clark

W. N. Papian

Appendix No. I General Programming of TX-0

II Proposed Conversion Program

III Direct Input Routine

APPENDIX I

## GENERAL PROGRAMMING OF TX-0

The TX-0 computer has an eighteen (18) bit word length. In the normal mode of operation it has four instructions and 4096 words of storage.

The four basic instructions in the normal mode are:

| $IR_0$ | $IR_1$ | CODE | DESCRIPTION |
|---|---|---|---|
| 0 | 0 | sto X | Store the contents of the accumulator in register X, leaving the contents of the accumulator unchanged. The previous contents of register X are destroyed. |
| 0 | 1 | add X | Add the contents of register X to the contents of the accumulator leaving the sum in the accumulator. The contents of register X are left unchanged. |
| 1 | 0 | trn X | Transfer on negative: If bit "0" of the accumulator is a "one" take the next instruction from register X and proceed from there. If bit "0" of the accumulator is a "zero" this instruction is ignored. |
| 1 | 1 | opr X | Operate class commands are done according to the number X. |

| IR | | | | | | MAR | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | | 3 | 4 | | 5 | 6 | 7 | 8 | 9 | 10 | | 11 | 15 | 12 | 13 | 14 | 16 | 17 | | | | |
| 00 | ST | 0.8 | ACL | 0.8 | ACR | 0.8 | 10 | IOS | 0.8 | 111 | P7H | 1.3 | 01 | MBR+IR | 1.1 | 10 | PEN | 1.2 | COMP | 1.4 | PAD | 1.7 | CRY | 1.2 | 01 | AMB |
| 01 | AD | | | | | 1.8 | 11 | HALT | 0.8 | 110 | P6H | 1.4 | 10 | SHR | 1.1 | 01 | TAC | | | | | | | 1.2 | 11 | TBR |
| 10 | TN | | | | | | | | 0.8 | 100 | PNT | 1.4 | 11 | CYR | | | | | | | | | | 1.3 | 10 | LMB |
| 11 | OP | | | | | | | | 0.8 | 001 | R1L | | | | | | | | | | | | | | | |
| | | | | | | | | | 0.8 | 011 | R3L | | | | | | | | | | | | | | | |
| | | | | | | | | | 0.8 | 010 | DIS | | | | | | | | | | | | | | | |
| | | | | | | | | | 0.8 | 101 | | | | | | | | | | | | | | | | |

AMB-Transfer the contents of AC to MBR.
TBR-Transfer the contents of TBR to MBR.
LMB-Live register to MBR.

CARRY

ADD-(partial add) Comp AC 1 if MBT1 1

COMP - Complement every digit of AC

PEN - Reserved for Light Pen.
TAC - Transfer the contents of the test accumulator register to the accumulator.

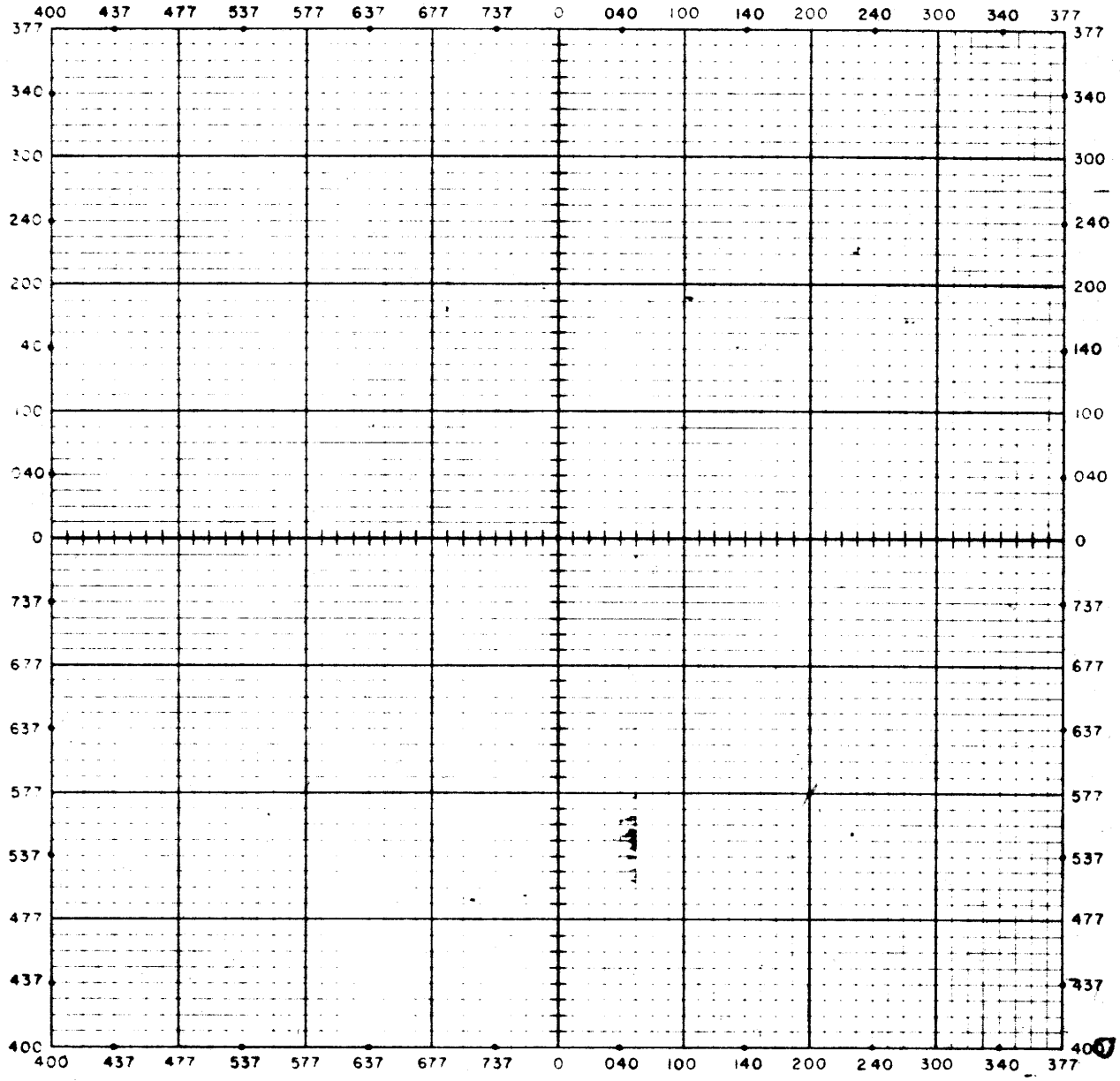
SHR - Shift AC right 1 place.
CYR - Cycle AC right 1 place.
MBL - Transfer contents of MBR to Live Register

P6H - Punch holes 1 thru 6 in Flexo Tape spec. By AC digits 2,5,8,11,14 & 17
P7H - Same as above but punch 7th hole continually.
PNT - Print one Flexo character spec. by AC digits 2,5,8,11,14 and 17.
R3L - Execute the double command CYR-R1L three times.
R1L - Read 1 line (6 holes of tape via PETR) to digits 0,3,6,9,12 & 15 of AC
DIS - Intensify a point on the scope with X-Y coordinates specified by AC.
101 - Pulse for EPSCO

IOS - Stop machine so that an in-out order specified by MAR 6,7,8 may be executed.
HALT - Halt the Computer

ACR - Clear right 9 bits of the Accumulator.

ACL - Clear left 9 bits of the Accumulator




TX-0 OCTAL GRAPH CHART

Monroe Weinstein
July 21, 1958

# APPENDIX II

## I DEFINITIONS

(a) Title = All characters before 1st ↵

(b) Comments = All characters after ↑ and before ↓

(a) If title ends in upper case, all characters before ↓

(b) Caution: →| and ↵ which are not affected by case shifts as far as printing is concerned will be ignored in upper case.

(c) Symbols = 1, 2, or 3 characters at least one letter and all either letters or digits

(d) Pseudo-instructions = 4 or more characters, at least one of the first 3 being a letter and all either letters or digits

(e) Integers = Sting of digits, interpreted as decimal or octal according as the last pseudo-instruction of the set (octal, decimal), and taken modulo $2^{18}-1$. i.e.,

$$+262143_{10} = +777777_8 = -0$$
$$-262143_{10} = -777777_8 = +0$$

## II SYLLABIES

Pseudo-instructions, symbols, and integers are syllables. One or more syllables, separated by plus or mimus signs, form a word. There are 5 types of words.

(a) Polysyllabic words without pseudo-instruction syllables.

1. Storage words terminated by →| or ↵

2. Address assignments terminated by ¦

(b) Monosyllabic symbolic words.

1. Symbolic address assignments terminated by (,)

2. Symbolic parameter assignments terminated by =(and followed by a storage word which is the equivalent and does not (immediately) enter storage).

(c) Pseudo-instruction words terminated by →| or ₂

III SYMBOL ASSIGNMENTS

The assembly program acts as if the following immediately followed the title:

| | | | |
|---|---|---|---|
| cll = 100000 | sto = 0 | add = 200000 | trn = 400000 |
| opr = 600000 | cll = 100000 | clr = 40000 | ios = 20000 |
| hlt = 30000 | p7h = 7000 | p6h = 6000 | pnt = 4000 |
| rlc = 1000 | r3c = 3000 | dis = 2000 | shr = 400 |
| cyr = 600 | mlr = 200 | pen = 100 | com = 40 |
| pad = 20 | cry = 10 | tac = 4 | tbr = 3 |
| lmb = 2 | amb = 1 | | |

An attempt to reassign a symbol by means of a symbolic address assignment will cause an alarm. However, symbols may be reassigned without alarm by means of the equal sign. Any symbol which has not been defined at the time it is needed to evaluate a word on the second pass will cause an alarm. To some extent it is possible to define symbols in terms of symbols. On the first pass a symbol assigned in terms of integers or previously defined symbols is defined. A symbol which is indefinite at the beginning of the second pass will not cause an alarm if its first appearance is an assignment in terms of symbols which are definite.

IV PSEUDO-INSTRUCTIONS

In addition to "octal" and "decimal", there are

(a) "start" the next word should be add + $x_2$ or trn + $x_2$ where x is the desired starting address. After this

word is read the pass is complete. After pass 1 the flexo tape should be replaced in the PETR and the "start" button pressed to do pass 2 provided there was no alarm on pass 1. The title and either "PASS 1" or "PASS 2" will print at the beginning of each pass so that the operator may verify that the right tape is reading.

(b) "binary" The 3 lines of tape immediately following are read in as a binary word and counted as a syllable for polysyllabic words thus 3-binary abc + 5 ב yields
$3 - 110_8 - 220020_8 - 044400_8 + 5 = -264520_8 = 513257_8$

(c) "call" Used as call + x,y it is translated as

```
        opr + cll + clr
        add + α
        trn + x
    α,trn + α + y
```

(c) "multiply" Used as multiply + x it is translated as

```
        opr + amb + mlr + pad
        add + α
        trn + mult
    α,trn + α + 2
        add + x
        opr + cll + clr + lmb + pad
```

and an indicator is set to provide a multiply subroutine

(e) Similar to "multiply" there may be
"divide", "square root", "sine", "cosine", etc.

SAMPLE PROGRAM

The next page is a program example.

```
 Program example
200| p,sto+p1
add+p2
sto+p3
prt=opr+pnt+cyr+ios
cla=opr+cll+clr
cla
p3,0       BECOMES ADD
prt
prt
prt+amb+com+pad
p1,0       BECOMES TRN
p2,add-trn-1
s,call+p,2            ENTER PRINT
binaryyes             PRINT YES
add+a
multiply+b
sto+c      (A)(B) TO (C)
opr+hlt
a=300
a| octal
300
b,5432
c,0
start+s
```

Charles Woodward
July 25, 1958

APPENDIX III

## DIRECT INPUT ROUTINE

Utility programs for the TX-0 are in roughly two groups. First there is a conversion or assembly program to convert programs punched on flexowriter tapes, either in absolute-address or symbolic address form, to tapes punched in the binary form the computer uses for operation. The conversion program uses all the memory; any tape produced by it must be subsequently run into the machine for operation.

The second group consists of a collection of programs which can coexist in the memory with operative programs. These are useful in debugging programs that have been written. The most important of these programs is Utility Tape 3, which provides a variety of printing and punching services. "Trace" is a program which is used in conjunction with Utility Tape 3 to follow the sequency of operations in a program being debugged. And finally, "Punchy" is a program which provides a somewhat more elaborate punch-out system than does Utility Tape 3.

### UTILITY TAPE 3

Utility Tape 3 is a program which provides for the TX-0, within the confines of the 4096-word memory, a direct-input utility system. The program features direct communication between the computer and the operator by means of the console Flexowriter.

Utility Tape 3 has a routine for examination and modification of the contents of any register at the Flexowriter keyboard. It has routines to search the entire memory and print out any registers with a specified contents or which refer to a specified address. It has a routine to compare a tape with the same program as it exists in

memory and to print out any changes that have occurred. It has a printing routine to print out any specified block of registers. And finally, it has a primitive routine to punch out the contents of any specified block of registers. UT3 occupies 1184 registers.

CONVERSION PROGRAM

Utility Tape 3 is always ready for conversion-program requests typed on the Flexowriter, except when some other specific routine, such as print-out, is in process.

The conversion program enables the operator to request a print-out of the contents of any specific register. The contents are printed out as instructions--a three-letter mnemonic symbol for any operate-class command for which such a symbol has been defined (Attachment III), three-letter command symbol followed by an octal address for any other instruction. If desired, UT3 will convert this to an octal constant. Any revision to the contents may be typed in as an octal number, a three-letter mnemonic, an addressed instruction with three-letter command symbol, or as any combination of mnemonics and octal constants separated by plus or minus signs.

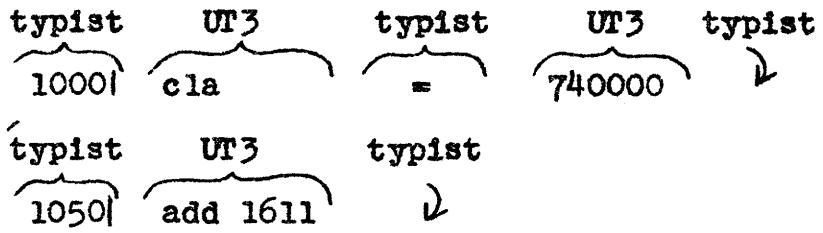
The contents of the register may be punched out in standard input-routine form, either in the original condition or as modified.

Register Examination

To examine the contents of a register, type the desired address followed by a slash. The utility program will print the contents as an instruction. If, after the instruction printout, a printout of the same register contents as an octal constant is desired, type an equal sign. UT3 will print out the contents as a constant and wait for further requests. In either case, if no

modification of the register contents is desired, terminate the line with a carriage return.

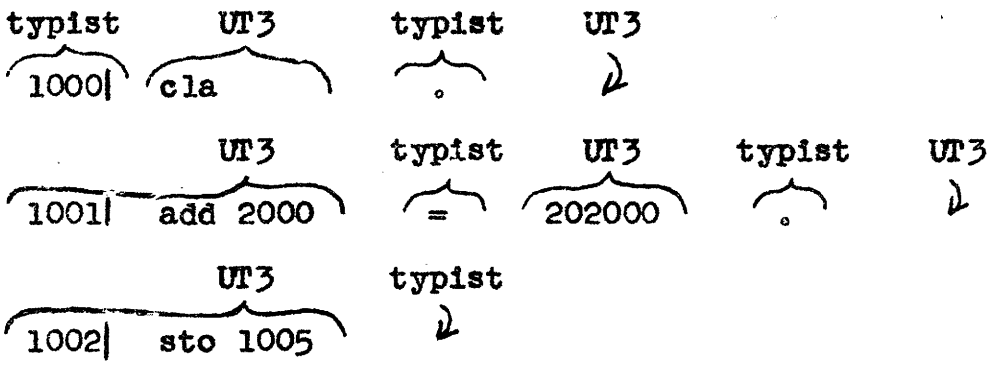
Example: Examine the contents of registers 1000 and 1050.




Examination of a Series of Registers

It is not necessary to type every address when examining or modifying two or more successive registers. After one register has been examined, the following one can also be examined by terminating the line with a period rather than a carriage return. The period tells UT3 to print a carriage return, the next address, a slash, and the contents of the next register.

Example: Examine registers 1000 to 1002.




Modifications

To modify the contents of a register, request the register for examination and after UT3 prints out the contents (as an instruction or as both an instruction and a constant) type the modification as an octal constant, three-letter symbol, octal-addressed instruction, or any combination of these separated by plus or minus signs. Follow

the typed modification by a carriage return. If you wish to examine the next register, terminate with a period rather than a carriage return. If you wish the modification to be punched out, terminate with a comma.

Example: Change the contents of register 1000 to clc, and the contents of 1001 to add 3000-5, that is, add 2773.

typist UT3 typist UT3

1000| cla clc. ↓

UT3 typist

1001| add 2000 add 3000-5 ↓

Modification Punch

To punch out the contents of a register on paper tape, follow the examination of that register with a comma. The utility system will punch out, in a form which can be read by the input routine, the contents of the register and then print a carriage return. If a modification is typed in before the comma UT3 will make the modification and punch out the modified contents.

Example: Punch out the change of register 1000 from cla to clc.

typist UT3 typist UT3

1000| cla clc, ↓

Transfer Control

To transfer control to some other program in the memory type go to, the register to which control is to be transferred, and a carriage return. UT3 transfers control to that register, entering with -0 in the accumulator.

Example: Transfer control to register 1000.

typist

go to 1000↓

## MEMORY-SEARCH PROGRAM

The memory-search program will search the entire memory to find all registers containing a specified word or all registers with contents referring to a specified address.

### Word Search

To search for a word, type word, the desired word and a carriage return. The word may be typed as an octal constant, an octal-addressed command, a three-letter operate-class symbol, or any combination of these separated by plus and minus signs. The routine searches the entire memory and prints out the address and contents of every register containing the desired word. A carriage return is printed when the search is finished. Search time is about 1 second.

### Address Search

To search for a reference to an address, type address, the desired address and a carriage return. The address may be typed as any combination of octal constants separated by plus and minus signs. The routine searches the entire memory for all words whose last 12 bits correspond to the given address. It will print the locations of such words, and the contents of the registers. A carriage return is printed when the search is finished.

The routine does not distinguish operate-class commands. For example, a request for references to address 40 will cause a print out of 1000 com (opr 40) and all other commands which complement the accumulator, in addition to valid references to register 40 (such as 1001 add 40).

## PROGRAM CHANGE DETECTION

To compare a program as it exists in memory with its original

form on tape, place the tape in the tape reader and type surprise and a carriage return. Surprise will read in the tape and then type out the location of any register whose present contents differ from the contents of the tape, the contents of that register, and then, in different color, the final tape contents for that register. The final contents may be the value produced by some modification at the back of the tape. The final tape contents is placed in the register by Surprise.

If the number of changes is greater than 39, Surprise will read in only part of the tape, print out the 39 changes, and then continue reading in the rest of the tape. If there should be a sum error in the read-in Surprise will type "error" and return control to UT3. The program in memory will not be disturbed unless Surprise had to read in the block containing the error in two or more sections.

PRINTING ROUTINES

Horizontal Print-Out

To print out the contents of a block of registers in horizontal format, type pnt horizontal (or just pnth), the first address of the block to be printed, and a carriage return. The print routine will answer by printing "to". The operator should then type the last address of the block and a carriage return. The routine will answer with "as". The operator may reply with either instructions or constants, and a carriage return.

The printing routine will then type out the specified block, 8 registers (octal 10) to a line. The address of the first register in each line will be printed at the left margin. If the first specified register is not an integral multiple of 10 (octal) a number of spaces will be left so that subsequent lines start with a multiple of 10.

If the request was for constants the printing will be as unsigned octal constants. If the request was for instructions the printout will be as three-letter mnemonics for all those commands for which the symbols have been defined and as the first two letters of the command symbol (i.e., st for sto, ad for add, tr for trn, and op for opr) followed by an octal address for all other commands.

Vertical Print-Out

To print the contents of a block of registers in vertical format type pnt vertical (or pntv), the first address of the block to be printed, and a carriage return. The machine replies and typing instructions are thereafter the same as for pnt horizontal.

Pntv will type out the specified block in two columns, 64 (octal 100) registers to a column. All pages are laid out so that the lead register in each column has an address which is some multiple of octal 100. If the specified first address of the block is not a multiple of 100, the routine will type a lead address and then space down enough lines to achieve uniform page layout before printing the first register.

Pntv starts by printing two carriage returns and the lead address, so to start printing at the top of one page, set the carriage at the last ruled line at the bottom of the previous page before typing the carriage return that gives control to pntv.

If constant printout was requested the contents of all registers will be printed as unsigned octal constants. If instruction printout was requested the routine will print three-letter mnemonic symbols for all operate-class commands for which such symbols have been defined, and full three-letter commands followed by octal address for all other commands.

## PUNCH-OUT ROUTINES

This program does not provide a title, and it does not punch out blocks in read-in-mode format -- only in normal input-routine format. It gives no choice of tape-termination commands, that is, starting commands -- it punches out only the type command that stops the computer after a tape is read in.

### Blank Tape

To feed six inches of blank tape, type a backspace.

### Input Routine

To punch out the input routine (Appendix I), type input and a carriage return. The input routine, followed by some blank tape, will be punched out in read-in-mode format. This input routine must be placed at the front of each tape.

### One-Word Block

To punch out a one-word block, use the comma as discussed in section on the conversion program.

### Block-Punch

To punch out a block of any length, type punch, the first address of the block to be punched, and a carriage return. The routine will reply with "to". Follow this with the last address and a carriage return. The routine will punch out the specified block in a form which can be read by the input routine, and then feed some blank tape.

### Starting Command

To punch out a start command, type start and the starting address. The routine will punch out the starting command in a form which will stop the computer after the tape is read in, that is, as an add command. For example, start 1000 will be punched out as add 1000.

## GENERAL INFORMATION ON UTILITY TAPE 3

### Operating Instructions

To use Utility Tape 3, first run in the tape bearing that title. The console Type In switch must be on and the Flexowriter Punch On switch depressed. If the Flexowriter power has been turned off it is necessary to "prime" it before typing by pressing the Start Read switch on the Flexowriter.

### Typing Instructions

"Listen", the subroutine which handles the Flexowriter communication, ignores all "invisible" characters (space, tab, stop code, delete, color change, upper case, lower case). It distinguishes between the letter l and the numeral 1; the two are not interchangeable. Listen has a single-x erasure -- typing one or more x's "erases" from memory all that has been typed in.

Computer words can be typed in as sums and differences of constants and commands. It is important to realize, however, that the sum of two operate-class commands is their numerical sum, and not the command for simultaneous performance of the two operation. For example, prt + cyl is not opr 27031, print and cycle left, but is 624000 + 600031 = 424032 or trn 24032.

UT3 types in the color opposite that used by the typist.

If some illegal combination is typed in, the routine will reply with "error" and wait for the correct answer. It should be pointed out that most safety checks had to be discarded in the interest of a short program, so it isn't hard to fool the system. For example, if at any time -- say when replying to the punch program -- the operator types instructions, the machine will immediately go to the print routine and do its best to print out some registers somewhere.

"Listen" looks only at the third and fourth characters of the word typed in to determine to what subroutine it should go. Therefore it is necessary to type only the first four letters of an answer, i.e., addr for address, pntv for pnt vertical, etc.

In order to shorten the program, UT3 frequently uses the live register for temporary storage. Since the live register is the Flexowriter input register, typing may upset the program in progress. An attempt was made to use the live register only where the introduction of spurious information will produce immediate and obvious results. However, it is a good idea to refrain from typing when any routine is in process.

Vocabulary

No provision has been made for enlarging the present vocabulary of mnemonic symbols for operate-class commands. It was thought that if additions were desired some of the less useful commands could be deleted.

The vocabulary is stored in registers 7432 to 7547, with binary and Flexowriter versions in alternate registers.

Program Location

Utility Tape 3 occupies the 1184 registers at the back of the 4096-word memory, from address 5540 through 7777 (octal). The routine is entered at register 5600. Registers 7612 through 7621 are not used by UT3 and may be used with any other program.

Flexowriter Setup

The printing routines of UT3 are designed to give correct layout for the following setup:

Left Margin: 8

Tabs: 16,26,36,46,56,66,76,86

Right Margin: 93

Subroutines

Utility Tape 3 was written as tightly as possible, so that very few general-purpose subroutines are available -- most subroutines have fixed entrances and exits, and rather peculiar characteristics. A few of the subroutines might prove useful in other programs, and so instructions for their use are provided.

Linkage

The TX-0 has no equivalent to the Whirlwind "A" register, so it is necessary to provide the return address to the subroutine by means of the contents of the accumulator when control is transferred to the subroutine. This is achieved by placing in the accumulator trn to the desired return address and then transferring to the subroutine, which stores this transfer command as an exit instruction.
Example: Transfer to Listen (register 5602) and return to 103.

```
100  cla
101  add 200
102  trn 5602
103  . . . program following the Listen subroutine
. . .
200  trn 103
```

All the subroutines listed in this memo return to the main program with -0 in the accumulator.

Constant Print-Out Subroutine

To print out an octal constant:

1. Store the word to be printed in register 5724.

2. Transfer control to register 6256 with the return address in the accumulator.

The word will be printed as an unsigned octal number. Initial zeros are suppressed except for the number +0, which will be printed out as 0. The contents of register 5724 will be destroyed in the print-out process.

Instruction Print-Out Subroutine

To print out a command:

1. Store the word to be printed in register 5724.

2. Transfer control to register 6172 with the return address in the accumulator.

The word stored will be printed as a three-letter mnemonic symbol or as a three-letter command followed by an octal address. The contents of register 5724 may be destroyed in the process.

Punch-Out Subroutine

To punch out the contents of a block of registers in normal-input-routine format:

1. Store the first address of the block in register 6432.

2. Store the last address of the block in register 7777.

3. Transfer control to register 7200 with the return address in the accumulator.

Flexowriter Communication

To use "Listen", the Flexowriter communication routine, transfer control to register 5602 with the return address in the accumulator.

Listen will store the characters typed in -- up to, but not including, a carriage return -- in "print-out" form, e.g. 110100 for g, in registers 7622 to 7741. The typed-in characters are followed by -0's. There are a number of exceptions to the "print-out" form of the stored characters. First of all, the "invisible"

characters (space, color change, stop code, upper case, lower case, tab, delete) and the letter z are not stored at all. The stored constants for the numerals 8 and 9 and the letters j, k and q do not correspond to "print-out" form. The plus sign is represented by the constant 200001, the minus sign by 300001. Finally, an x "erases" all that has been typed in.

Care must be used in typing when using Listen -- typing any transfer-control symbol or command, such as a slash, period, comma, equal sign, back space, or words such as start will transfer control to the appropriate portion of the utility system.

To obtain the binary form of a typed-in octal number, mnemonic symbol, command, or any combination of these separated by plus or minus signs, use "Listen" and then "Interpret". Enter the "Interpret" subroutine at register 5720 with the return address in the accumulator. The answer will appear in register 7777.

Example: The binary form of a typed-in number is to be obtained.

```
1001  cla
1011  add  200
1021  trn 5602
1031  add  201
1041  trn 5720
1051  cla
1061  add 7777

2001  trn  103
2011  trn  105
```

ATTACHMENT I

## THE INPUT ROUTINE

The input routine is discussed in 6M-4789. However, since the existing input routine is shifted downward 170000 registers from the large-memory routine it was thought advisable to include a second description here.

A read-in mode is provided in the TX-0 which makes it possible to read in tapes punched out with sto x instructions alternating with the word which is to go in register x. However, because of the store command which must accompany each word, such tapes become quite long and furthermore there is no check of read-in errors, so the usual procedure is to place on the front of every tape an input routine, to be read in read-in mode, and, when this is in, transfer control to it so the rest of the tape can be read in a shorter format. A printout of the input routine is included. A block of a tape to be read in by this routine has as its first word a command to store in the first address, as its second word the complement of last address, then the block of contents for the registers indicated, and, at the end, the complement of the sum of everything previously printed out in that block, including the two addresses.

If the sum formed by the input routine agrees with the sum punched on the tape the routine will proceed to read the next block. If the sum does not agree the machine will halt in register 7772.

A tape read in by the input routine can be terminated in either of two ways. If the last block on the tape is followed by a transfer instruction (say trn 1000) read-in will stop and control will be transferred immediately to the register indicated (that is, 1000). If the last block is followed by an add command (say add 1000) the computer will halt in register 7776 and when the restart button is pushed will transfer control to the indicated register (that is 1000).

# INPUT ROUTINE

```
7742 add 7773
7743 trn 7772
7744 r3c          ENTER HERE
7745 sto 7756
7746 trn 7756
7747 add 7774
7750 trn 7775
7751 r3c
7752 sto 7777
7753 add 7756
7754 sto 7741
7755 r3c
7756 add 5600
7757 add 7741
7760 sto 7741
7761 cla
7762 add 7756
7763 add 7773
7764 sto 7756
7765 add 7777
7766 trn 7755
7767 r3c
7770 add 7741
7771 trn 7742
7772 hlt          HALT AFTER SUM ERROR
7773 sto 1
7774 add
7775 sto 7777
7776 hlt          HALT AFTER READ IN
```

## ATTACHMENT II

### THE BASIC OPERATE-CLASS COMMANDS*

| Mnemonic Symbol | Command | Bit(s) | Time Pulse | Description |
|---|---|---|---|---|
| cll | opr 100000 | 2 | 0.8 | clear left 9 bits of AC |
| clr | opr 40000 | 3 | 0.8 | clear right 9 bits of AC |
| --- | opr 20000 | 4 | 0.8 | in-out-stop |
| hlt | opr 30000 | 4,5 | 1.8 | halt |
| --- | opr 7000 | 6,7,8 | 0.8 | punch 7 holes |
| --- | opr 6000 | 6,7 | 0.8 | punch 6 holes |
| --- | opr 4000 | 6 | 0.8 | print |
| --- | opr 1000 | 8 | 0.8 | read 1 line |
| --- | opr 3000 | 7,8 | 0.8 | read 3 lines |
| --- | opr 2000 | 7 | 0.8 | display on scope |
| shr | opr 400 | 9 | 1.4 | shift right |
| cyr | opr 600 | 9,10 | 1.4 | cycle right |
| lro* | opr 200 | 10 | 1.3 | (MBR)→ (LR)* |
| --- | opr 100 | 11 | 1.1 | read light pen |
| --- | opr 4 | 15 | 1.1 | TAC ones → (AC) |
| com | opr 40 | 12 | 1.2 | complement (AC) |
| --- | opr 20 | 13 | 1.4 | partial add (MBR) and (AC) |
| --- | opr 10 | 14 | 1.7 | partial add carry digits (AC) |
| --- | opr 1 | 17 | 1.2 | (AC) → (MBR) |
| --- | opr 3 | 16,17 | 1.7 | (TBR) → (MBR) |
| --- | opr 2 | 16 | 1.3 | (LR) → (MBR) |

* This table is adapted from Appendix B, 6M-4789, taking into consideration changes in mnemonic symbols since the publication of that memo.

* Note that if this is used alone it clears the live register, since the MBR is empty.

## ATTACHMENT III

### STORED CONSTANTS

A number of commonly used constants are included in Utility Tape 3 and need not be provided again for each individual program.

| Constant | Location |
|---|---|
| 1 | 7773 |
| 2 | 7373 |
| 10 | 7334 |
| 100 | 7563 |
| -0=777777 | 7342 |
| -1=777776 | 7423 |
| -2=777775 | 7413 |
| -100=777677 | 7352 |
| add=200000 | 7774 |
| trn=400000 | 7437 |
| opr=600000 | 7441 |
| add 1=200001 | 7347 |
| -add=577777 | 7336 |
| -add-1=577776 | 7567 |
| carriage return=101001 | 7335 |
| tab=100101 | 7420 |

These constants are not affected by reading in the Trace program, but are destroyed by Punchy.

TRACE

Trace is a utility program which makes it possible to follow the path of an operating program. It operates as an interpretive routine and periodically prints out all addresses from which the computer transferred control on a trn instruction.

Operating Instructions

Utility Tape 3 must be in the machine. Run in Trace. This will destroy the pnt vertical and pnt horizontal programs of UT3.

To follow the path of a program type trace and a carriage return. Follow this with begin, the starting address of the program, and a carriage return.

The program operations will be performed in an interpretive fashion until 79 transfers have been performed, that is, until the program has arrived at 79 trn instructions with a negative accumulator. Trace will then print "from" and the addresses from which control was transferred, that is, the locations of the 79 trn instructions. It will then resume interpretive operation of the program until 79 more transfers have been performed.

If the program contains an hlt command (or any other command which would in normal operation stop the computer) trace will type "from" and all addresses from which it transferred control. It will then type "halt" and the contents of the program counter, accumulator and live register when the hlt instruction was reached. Control is then returned to Utility Tape 3.

In many cases the operator is interested in only a few transfer instructions and the complete printout is needlessly time consuming. It is possible to restrict the printout to a specified range of registers. To do this, type trace, the first register of

interest, and a carriage return. Answer the machine reply "to" with the last register of interest and a carriage return. Then type begin, the starting address of the program and a carriage return. Trace will still store all addresses from which it transfers control, but will print out only those in the specified block.

Program Speed

Trace reduces the average speed of operation of the machine by a factor of 15 to 20, plus printing time, which may make it unusable for some applications. The actual times are given by the table below.

| Command | Machine operations when using trace |
|---|---|
| sto | 13 |
| add | 13 |
| trn (positive accumulator) | 13 |
| trn (negative accumulator) | 22* |
| opr | 22 |

* plus a pause for print-out after 79 transfers

PUNCHY

The advantages of Punchy over the punch routine of Utility Type 3 are: (1) title-punching facilities, (2) optional read-in mode, and (3) a choice of tape termination commands.

If the restart button is pushed after Punchy is read in, Punchy will feed some blank tape and type "title is". The operator will then type in the desired title of the tape, followed by a carriage return. The title will be punched out in six-hole form on the tape. If no title is desired the operator can type a tab and a carriage return.

After the title is complete Punchy will type "input routine layout". If normal input routine layout of the tape is desired the operator should type yes and a carriage return. If read-in mode layout is desired he should type no. If the answer is yes Punchy will punch out the input routine in read-in mode format at the front of the tape.

Punchy will next say "first address" which should be answered by typing the desired first address. Punchy then will say "last address" which should be answered with the desired last address of the block to be punched. Upon receiving the last address Punchy will punch the desired block and then type "more blocks". If there are more blocks to be punched the answer is yes; Punchy will ask for the addresses again. If there are no more blocks, the answer is no. Punchy will then type "start at" which should be answered by the desired starting address. Punchy will reply with "stop after read-in". A yes answer will cause the starting command to be punched out so as to halt the computer after the tape is read in (an add command). A no answer will cause it to be punched out

so to transfer control directly to the indicated register (a trn command). After punching out the starting command and feeding blank tape the computer will halt. Restarting will cause Punchy to start all over again.

Punchy Locations

Punchy occupies 665 (decimal) registers at the back of the small memory, addresses 6547 to 7777. The starting address is 6550. Punchy and Utility Tape 3 occupy some of the same registers, and so cannot coexist in the memory.

Typing Instructions

Typing communication with Punchy is quite straightforward. Punchy ignores spaces, except when the title is being typed, so that inadvertent spaces cause no trouble. If an error is made in typing the operator can type xx (or any number of x's greater than two) and "erase" from the memory all that has been typed. If the operator gives an illegal reply to a Punchy request, Punchy will print out "error" and wait for a corrected answer.

In determining if an answer is yes or no Punchy examines only the first letter typed in, so that one can get by with just typing y or n.

Charles Woodward
July 25, 1958

# APPENDIX II
## DIRECT INPUT ROUTINE

Utility programs for the TX-0 are in roughly two groups. First there is a conversion or assembly program to convert programs punched on flexowriter tapes, either in absolute-address or symbolic address form, to tapes punched in the binary form the computer uses for operation. The conversion program uses all the memory; any tape produced by it must be subsequently run into the machine for operation.

The second group consists of a collection of programs which can coexist in the memory with operative programs. These are useful in debugging programs that have been written. The most important of these programs is Utility Tape 3, which provides a variety of printing and punching services. "Trace" is a program which is used in conjunction with Utility Tape 3 to follow the sequence of operations in a program being debugged. And finally, "Punchy" is a program which provides a somewhat more elaborate punch-out system than does Utility Tape 3.

### UTILITY TAPE 3

Utility Tape 3 is a program which provides for the TX-0, within the confines of the 4096-word memory, a direct-input utility system. The program features direct communication between the computer and the operator by means of the console Flexowriter.

Utility Tape 3 has a routine for examination and modification of the contents of any register at the Flexowriter keyboard. It has routines to search the entire memory and print out any registers with a specified contents or which refer to a specified address. It has a routine to compare a tape with the same program as it exists in memory and to print out any changes that have occured. It has a printing routine to print out any specified block of registers. And finally, it has a primitive routine to punch out the contents of any specified block of registers. UT3 occupies 1184 registers.

### CONVERSION PROGRAM

Utility Tape 3 is always ready for conversion-program requests typed on the Flexowriter, except when some other specific routine, such as print-out, is in process.

The conversion program enables the operator to request a print-out of the contents of any specific register. The contents are printed out as instructions--a three-letter mnemonic symbol for any operate-class command for which

such a symbol has been defined (Attachment III), three-letter command symbol followed by an octal address for any other instruction. If desired, UT3 will convert this to an octal constant. Any revision to the contents may be typed in as an octal number, a three-letter mnemonic, an addressed instruction with three-letter command symbol, or as any combination of mnemonics and octal constants separated by plus or minus signs.

The contents of the register may be punched out in standard input-routine form, either in the original condition or as modified.

Register Examination

To examine the contents of a register, type the desired address followed by a slash. The utility program will print the contents as an instruction. If, after the instruction printout, a printout of the same register contents as an octal constant is desired, type an equal sign. UT3 will print out the contents as a constant and wait for further requests. In either case, if no modification of the register contents is desired, terminate the line with a carriage return.

Example: Examine the contents of registers 1000 and 1050.

| typist | UT3 | typist | UT3 | typist |
|---|---|---|---|---|
| 1000/ | cla | = | 740000 | ↵ |

| typist | UT3 | typist |
|---|---|---|
| 1050/ | add 1611 | ↵ |

Examination of a Series of Registers

It is not necessary to type every address when examining or modifying two or more successive registers. After one register has been examined, the following one can also be examined by terminating the line with a period rather than a carriage return. The period tells UT3 to print a carriage return, the next address, a slash, and the contents of the next register.

Example: Examine registers 1000 to 1002.

| typist | UT3 | typist | UT3 |
|---|---|---|---|
| 1000/ | cla | . | ↵ |

| | UT3 | typist | UT3 | typist | UT3 |
|---|---|---|---|---|---|
| 1001/ | add 2000 | = | 202000 | . | ↵ |

| | UT3 | typist |
|---|---|---|
| 1002/ | sto 1005 | ↵ |

## Modifications

To modify the contents of a register, request the register for examination and after UT3 prints out the contents (as an instruction or as both an instruction and a constant) type the modification as an octal constant, three-letter symbol, octal-addressed instruction, or any combination of these separated by plus or minus signs. Follow the typed modification by a carriage return. If you wish to examine the next register, terminate with a period rather than a carriage return. If you wish the modification to be punched out, terminate with a comma.

Example: Change the contents of register 1000 to clc, and the contents of 1001 to add 3000-5, that is, add 2773.

| typist | UT3 | typist | UT3 |
|---|---|---|---|
| 1000/ | cla | clc. | ↵ |

| UT3 | typist |
|---|---|
| 1001/ add 2000 | add 3000-5↵ |

## Modification Punch

To punch out the contents of a register on paper tape, follow the examination of that register with a comma. The utility system will punch out, in a form which can be read by the input routine, the contents of the register and then print a carriage return. If a modification is typed in before the comma UT3 will make the modification and punch out the modified contents.

Example: Punch out the change of register 1000 from cla to clc.

| typist | UT3 | typist | UT3 |
|---|---|---|---|
| 1000/ | cla | clc. | ↵ |

## Transfer Control

To transfer control to some other program in the memory type go to, the register to which control is to be transferred, and a carriage return. UT3 transfers control to that register, entering with -0 in the accumulator.

Example: Transfer control to register 1000.

| typist |
|---|
| go to 1000↵ |

## MEMORY-SEARCH PROGRAM

The memory-search program will search the entire memory to find all registers containing a specified word or all registers with contents referring to a specified address.

### Word Search

To search for a word, type word, the desired word and a carriage return. The word may be typed as an octal constant, an octal-addressed command, a three-letter operate-class symbol, or any combination of these separated by plus and minus signs. The routine searches the entire memory and prints out the address and contents of every register containing the desired word. A carriage return is printed when the search is finished. Search time is about 1 second.

### Address Search

To search for a reference to an address, type address, the desired address and a carriage return. The address may be typed as any combination of octal constants separated by plus and minus signs. The routine searches the entire memory for all words whose last 12 bits correspond to the given address. It will print the locations of such words, and the contents of the registers. A carriage return is printed when the search is finished.

The routine does not distinguish operate-class commands. For example, a request for references to address 40 will cause a print out of 1000 com (opr 40) and all other commands which complement the accumulator, in addition to valid references to register 40 (such as 1001 add 40).

## PROGRAM CHANGE DETECTION

To compare a program as it exists in memory with its original form on tape, place the tape in the tape reader and type surprise and a carriage return. Surprise will read in the tape and then type out the location of any register whose present contents differ from the contents of the tape, the contents of that register, and then, in different color, the final tape contents for that register. The final contents may be the value produced by some modification at the back of the tape. The final tape contents is placed in the register by Surprise.

If the number of changes is greater than 39, Surprise will read in only part of the tape, print out the 39 changes, and then continue reading in the rest of the tape. If there should be a sum error in the read-in Surprise will type

"error" and return control to UT3. The program in memory will not be disturbed unless Surprise had to read in the block containing the error in two or more sections.

## PRINTING ROUTINES

### Horizontal Print-Out

To print out the contents of a block of registers in horizontal format, type pnt horizontal (or just pnth), the first address of the block to be printed, and a carriage return. The print routine will answer by printing "to". The operator should then type the last address of the block and a carriage return. The routine will answer with "as". The operator may reply with either instructions or constants, and a carriage return.

The printing routine will then type out the specified block, 8 registers (octal 10) to a line. The address of the first register in each line will be printed at the left margin. If the first specified register is not an integral multiple of 10 (octal) a number of spaces will be left so that subsequent lines start with a multiple of 10.

If the request was for constants the printing will be as unsigned octal constants. If the request was for instructions the printout will be as three-letter mnemonics for all those commands for which the symbols have been defined and as the first two letters of the command symbol (i.e., st for sto, ad for add, tr for trn, and op for opr) followed by an octal address for all other commands.

### Vertical Print-Out

To print the contents of a block of registers in vertical format type pnt vertical (or pntv), the first address of the block to be printed, and a carriage return. The machine replies and typing instructions are thereafter the same as for pnt horizontal.

Pntv will type out the specified block in two columns, 64 (octal 100) registers to a column. All pages are laid out so that the lead register in each column has an address which is some multiple of octal 100. If the specified first address of the block is not a multiple of 100, the routine will type a lead address and then space down enough lines to achieve uniform page layout before printing the first register.

Pntv starts by printing two carriage returns and the lead address, so to start printing at the tope of one page, set the carriage at the last ruled line at the bottom of the previous page before typing the carriage return that gives control to pntv.

If constant printout was requested the contents of all registers will be printed as unsigned octal constants. If instruction printout was requested the routine will print three-letter mnemonic symbols for all operate-class commands for which such symbols have been defined, and full three-letter commands followed by octal address for all other commands.

## PUNCH-OUT ROUTINES

This program does not provide a title, and it does not punch out blocks in read-in-mode format -- only in normal input-routine format. It gives no choice of tape-termination commands, that is, starting commands -- it punches out only the type command that stops the computer after a tape is read in.

### Blank Tape

To feed six inches of blank tape, type a backspace.

### Input Routine

To punch out the input routine (Appendix I), type input and a carriage return. The input routine, followed by some blank tape, will be punched out in read-in-mode format. This input routine must be placed at the front of each tape.

### One-Word Block

To punch out a one-word block, use the comma as discussed in section on the conversion program.

### Block-Punch

To punch out a block of any length, type punch, the first address of the block to be punched, and a carriage return. The routine will reply with "to". Follow this with the last address and a carriage return. The routine will punch out the specified block in a form which can be read by the input routine, and then feed some blank tape.

### Starting Command

To punch out a start command, type start and the starting address. The routine will punch out the starting command in a form which will stop the computer after the tape is read in, that is, as an add command. For example, start 1000 will be punched out as add 1000.

## GENERAL INFORMATION ON UTILITY TAPE 3

### Operating Instructions

To use Utility Tape 3, first run in the tape bearing that title. The console Type In switch must be on and the Flexowriter Punch On switch depressed.

If the Flexowriter power has been turned off it is necessary to "prime" it before typing by pressing the Start Read switch on the Flexowriter.

Typing Instructions

"Listen", the subroutine which handles the Flexowriter communication, ignores all "invisible" characters (space, tab, stop code, delete, color change, upper case, lower case). It distinguishes between the letter l and the numeral 1; the two are not interchangeable. Listen has a single-x erasure -- typing one or more x's "erases" from memory all that has been typed in.

Computer words can be typed in as sums and differences of constants and commands. It is important to realize, however, that the sum of two operate-class commands is their numerical sum, and not the command for simultaneous performance of the two operation. For example, prt + cyl is not opr 27031, print and cycle left, but is 624000 + 600031 = 424032 or trn 24032.

UT3 types in the color opposite that used by the typist.

If some illegal combination is typed in, the routine will reply with "error" and wait for the correct answer. It should be pointed out that most safety checks had to be discarded in the interest of a short program, so it isn't hard to fool the system. For example, if at any time -- say when replying to the punch program -- the operator types instructions, the machine will immediately go to the print routine and do its best to print out some registers somewhere.

"Listen" looks only at the third and fourth characters of the word typed in to determine to what subroutine it should go. Therefore it is necessary to type only the first four letters of an answer, i.e., addr for address, pntv for pnt vertical, etc.

In order to shorten the program, UT3 frequently uses the live register for temporary storage. Since the live register is the Flexowriter input register, typing may upset the program in progress. An attempt was made to use the live register only where the introduction of spurious information will produce immediate and obvious results. However, it is a good idea to refrain from typing when any routine is in process.

Vocabulary

No provision has been made for enlarging the present vocabulary of mnemonic symbols for operate-class commands. It was thought that if

additions were desired some of the less useful commands could be deleted.

The vocabulary is stored in registers 7432 to 7547, with binary and Flexowriter versions in alternate registers.

Program Location

Utility Tape 3 occupies the 1184 registers at the back of the 4096-word memory, from address 5540 through 7777 (octal). The routine is entered at register 5600. Registers 7612 through 7621 are not used by UT3 and may be used with any other program.

Flexowriter Setup

The printing routines of UT3 are designed to give correct layout for the following setup:

Left Margin: 8

Tabs: 16, 26, 36, 46, 56, 66, 76, 86

Right Margin: 93

Subroutines

Utility Tape 3 was written as tightly as possible, so that very few general-purpose subroutines are available--most subroutines have fixed entrances and exits, and rather peculiar characteristics. A few of the subroutines might prove useful in other programs, and so instructions for their use are provided.

Linkage

The TX-0 has no equivalent to the Whirlwind "A" register, so it is necessary to provide the return address to the subroutine by means of the contents of the accumulator when control is transferred to the subroutine. This is achieved by placing in the accumulator trn to the desired return address and then transferring to the subroutine, which stores this transfer command as an exit instruction. Example: Transfer to Listen (register 5602) and return to 103.

```
100  cla
101  add 200
102  trn 5602
103  . . . program following the Listen subroutine
. . .
200  trn 103
```

All subroutines listed in this memo return to the main program with -0 in the accumulator.

Constant Print-Out Subroutine

To print out an octal constant:

1. Store the word to be printed in register 5724.

2. Transfer control to register 6256 with the return address in the accumulator.

The word will be printed as an unsigned octal number. Initial zeros are suppressed except for the number +0, which will be printed out as 0. The contents of register 5724 will be destroyed in the print-out process.

Instruction Print-Out Subroutine

To print out a command:

1. Store the word to be printed in register 5724.

2. Transfer control to register 6172 with the return address in the accumulator.

The word stored will be printed as a three-letter mnemonic symbol or as a three-letter command followed by an octal address. The contents of register 5724 may be destroyed in the process.

Punch-Out Subroutine

To punch out the contents of a block of registers in normal-input-routine format:

1. Store the first address of the block in register 6432.

2. Store the last address of the block in register 7777.

3. Transfer control to register 7200 with the return address in the accumulator.

Flexowriter Communication

To use "Listen", the Flexowriter communication routine, transfer control to register 5602 with the return address in the accumulator.

Listen will store the characters typed in -- up to, but not including, a carriage return -- in "print-out" form, e.g. 110100 for g, in registers 7622 to 7741. The typed-in characters are followed by -0's. There are a number of exceptions to the "print-out" form of the stored characters. First of all, the "invisible" characters (space, color change, stop code, upper case, lower case, tab, delete) and the letter z are not stored at all. The stored constants for the numerals 8 and 9 and the letters j, k and q do not correspond to "print-out" form. The plus sign is represented by the constant 200001, the minus sign by 300001. Finally, an x "erases" all that has been typed in.

Care must be used in typing when using Listen -- typing any transfer-control symbol or command, such as a slash, period, comma, equal sign, back space, or words such as start will transfer control to the appropriate portion of the utility system.

To obtain the binary form of a typed-in octal number, mnemonic symbol command, or any combination of these separated by plus or minus signs, use "Listen" and then "Interpret". Enter the "Interpret" subroutine at register 5720 with the return address in the accumulator. The answer will appear in register 7777.

Example: The binary form of a typed-in number is to be obtained.

```
100 | cla
100 | add  200
102 | trn  5602
103 | add  201
104 | trn  5720
105 | cla
106 | add  7777

200 | trn  103
201 | trn  105
```