1. IDENTIFICATION

1.1 Digital-8-35-S-B

1.2 680 8-Bit Character Assembly Subroutines

1.3 November 8, 1965
2. ABSTRACT

The 680 Data Communication System 8-Bit Character Assembly Subroutines concentrate Teletype data by assembling serial-bit data into 8-bit characters and present the user with data similar to that obtained by using a 630 DCS and scanner. They also add start and stop bits to 8-bit characters and transmit them in serial-bit fashion. Full duplex lines are assumed, but the subroutines will work with half duplex if the user handles the expected echo.

3. REQUIREMENTS

3.1 Storage

The subroutines as presently coded occupy 400 octal locations plus space for internal buffering of the input and output characters and for the TTI instructions. In addition, space is used in memory page 0 and a limited number of autoindex registers are used as explained below. Within the limits described, the program can be placed anywhere in the first 4K of PDP-8 memory. The total amount of memory used including the autoindex registers and the locations in page 0 is as follows:

\[422_8 + 7n\]

where \(n\) is the number of Teletype lines to the next even multiple of eight lines if the number of lines is not already an even multiple of eight.

3.2 Subprograms and/or Subroutines

Digital-8-35-S-A
680 5-Bit Character Assembly Subroutines

for reference or in the event the user's requirements include a mixture of 5-bit and 8-bit lines.

3.3 Equipment

Minimum configuration PDP-8
680 Data Communication System hardware

3.4 Miscellaneous

3.4.1 The tag TT8BGN must be defined as the address of the start of the Teletype subroutines. It can be defined as anywhere in memory, but must be equated to the start of a PDP-8 memory page.

3.4.2 Three autoindex registers called TBA1X, TBA2X, and TBA3X must be defined.

3.4.3 The tag TT8PG0 must be defined as the start of an area in memory page 0 where the necessary Teletype constants can be stored. An area of 178 registers must be reserved.

3.4.4 The tag T8OB2F must be defined as the start of the area reserved for outputting the Teletype characters. It must be equal in length to the number of lines (even multiple of 8) attached to the particular set of subroutines. It can be anywhere in memory and need not start at the beginning of a memory page.

3.4.5 The tag T8OB2F2 must be defined as an area equal in length to T8OB2F. It is used for double-buffering the output characters to allow maximum output rate.

3.4.6 The tag T8IB2F must be defined as the area for storing incoming Teletype characters and line numbers. It must be equal in length to twice the number of lines attached to the particular set of subroutines.
3.4.7 The tag T8IN must be defined as the start of the area used by the subroutines for generating the appropriate number of TTl instructions. It must be equal in length to three times the number of lines plus one register. Here again it need not be defined as the start of a memory page.

3.4.8 The tag TTCHAR must be defined as a single register in page O.

3.4.9 In the interrupt service routine the following set or sets of instructions must appear:

\[
\begin{align*}
\text{TBS KP} & \quad / \text{SKIP ON CLOCK FLAG} \\
\text{SKP} & \quad / \text{TEST FOR NEXT INTERRUPT CAUSE} \\
\text{JMP T8DIS} & \quad / \text{JUMP TO APPROPRIATE CLOCK INTERRUPT ROUTINE}
\end{align*}
\]

Because of the speed necessary for Teletype handling, the checks for clock interrupts should be the first ones in the interrupt service interrogation loop; the link bit and accumulator contents should not be saved prior to interrogation of the appropriate clock flag. If necessary for other interrupts, the link and accumulator contents should be saved only after all clock interrupts have been checked.

3.4.10 Clock IOT's

The IOT's to test the clock for the 1 state, turn the clock on, and turn the clock off must be given the correct octal definitions:

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Clock 1</th>
<th>Clock 2</th>
<th>Clock 3</th>
<th>Clock 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBSKP</td>
<td>6421</td>
<td>6431</td>
<td>6441</td>
<td>6451</td>
</tr>
<tr>
<td>TT8ON</td>
<td>6424</td>
<td>6434</td>
<td>6444</td>
<td>6454</td>
</tr>
<tr>
<td>TT8OFF</td>
<td>6422</td>
<td>6432</td>
<td>6442</td>
<td>6452</td>
</tr>
</tbody>
</table>

4. USAGE

4.2 Calling Sequence

The pseudo command T8INIT must be executed before the instruction TT8ON and also before either of the other pseudo commands T8SOF or T8SIR is executed. (See Section 4.4.1, 4.4.2 and 4.4.3 for definitions of the pseudo commands.)

4.3 Switch Settings

None

4.4 Start up and/or Entry

Three pseudo commands for using this set of subroutines are provided to the main program. They are defined as jumps to subroutines and their definitions and instructions are included in the package. These are the only commands necessary in the main program for gathering and outputting the Teletype characters. The user should note that no subroutines are included for packing or unpacking of the characters by word or even line number.

4.4.1 Teletype Initialize (T8INIT)

This command (which must be used only once in the main program) assumes that the user enters with the number of lines in the accumulator and that the register following the initialize command
contains the first line number for this type of Teletype line. This subroutine initializes all of the buffer areas, counters, and pointers, and generates the proper number of TTI instructions.

### 4.4.2 Skip if Output Free (T8SOF)

This instruction skips the next register in memory and transmits the character contained in register TTCHAR if the indicated output line is free. If the output line is not free, the instruction does not skip. The instruction requires that the line number over which the character is to be transmitted be in the accumulator at the time the instruction is issued. The pseudo command takes 24 μsec minimum time, and 42 μsec maximum time. The accumulator will be cleared when exiting from the command.

### 4.4.3 Skip if Input Ready (T8SIR)

This instruction skips the next location in memory and returns with the line number in the accumulator and the character placed at TTCHAR if an input character is available. If no character is available, the instruction does not skip and the accumulator is -1. Only the low order eight bits of the character at TTCHAR should be used, as additional bits representing the stop codes are also present in the character.

If no character is available, 15 μsec are used by the pseudo instructions; if a character is available, 37.5 μsec are used; and if the end of the storage area is reached, a maximum of 48 μsec is used by the instruction.

### 4.4.4 Skip if Either Input is Ready (T8SIE)

This command is used when more than one type of Teletype line is being used. As presently coded, it assumes that two are being used, that one is an 8-bit line, and one is a 5-bit line. It skips if a character is available from either of the two types of lines being used and presents the user with the line number in the accumulator and the character in location TTCHAR. If no character is available, 37.5 μsec are used; if a character is available, 60 μsec are used; if the end of a buffer area is found, a maximum of 70.5 μsec is used. As stated, the command assumes that two types of lines are being used. However, if a single type of line at two different speeds is used, the instructions, which occupy approximately 10 locations, can easily be changed to reference the correct set of subroutines.

**NOTE:** Since the pseudo operation references the pseudo-operation T5SIR, the user can expect an assembly error if the 8-bit subroutines are not assembled with the 5-bit subroutines. However, assuming that the user's requirements include only 8-bit lines and that the operation T8SIE would not be used, the error may either be ignored or the coding for T8SIE be deleted from the ASCII tape.

### 5. RESTRICTIONS

#### 5.1 Status Active Registers

The autoindex registers defined as T8AX1, T8AX2, and T8AX3 must not be disturbed after the pseudo operation T8INIT.

### 6. DESCRIPTION

#### 6.1 Discussion

These subroutines are designed to accumulate 8-bit Teletype characters to and from multiple Teletype lines connected to a PDP-8. They handle input data in serial-bit format and present the user
with character and line identification. The user presents the routines with line identification and character format data, and the routines transmit the information in serial-bit format.

Most of the PDP-8 memory is available for data buffering and for packing. A large proportion of the time however is used in buffering the Teletype lines themselves. Assuming even minor data handling is necessary before transmission (possibly to a larger computer), present estimates indicate the user cannot handle 128 8-bit lines at 110 baud. Exact timing information is shown in Section 9. The user should note that the programming described involves the handling of the Teletype lines only and does not include any packing or unpacking of words, lines, or messages. The main program communicates with the Teletype subroutines via a group of pseudo commands which are described fully in Section 4.4 with examples of their usage in Section 6.2.

If the user's requirements include a mixture of 8-bit and 5-bit lines, it is necessary that the 5-bit Character Assembly Subroutines (Digital-8-35-S-A) be included with the user's programs.

6.2 Examples and/or Applications

6.2.1 To initialize the subroutines, coding similar to the following should appear in the user's program:

```
TAD NUMLIN  // GET NUMBER OF LINES
TBINIT     // INITIALIZE SUBROUTINES
SLN        // STARTING LINE NUMBER
ION        // ENABLE INTERRUPTS
TTBON      // TURN ON CLOCK
```

NOTE: Following these lines of coding it is necessary that the user wait 8 clock interrupts before using the pseudo-instruction TBSOF. Otherwise, the first character transmitted will be erroneous.

6.2.2 To output a character, coding similar to the following should appear:

```
TAD CHARAC  // GET OUTPUT CHARACTER
DCA TTCHAR  // FOR OUTPUT SUBROUTINE
TAD LINE NO // GET LINE NUMBER
TBSOF       // OUTPUT, SKIP IF FREE
JMP OUTNA   // OUTPUT NOT FREE
CONTINUE    // CHARACTER ACCEPTED, CONTINUE
```

6.2.3 To test for an input character available, coding similar to the following should appear:

```
TBSIR       // CHECK FOR INPUT
JMP .-1     // WAIT FOR A CHARACTER
DCA SAVLIN  // SAVE LINE NUMBER
TAD TTCHAR  // GET CHARACTER INPUT
AND THREE7  // 377, CLEAR STOP BIT
```

7. METHODS

7.1 Discussion
7.1.1 Input Character Assembly

The 8-bit Character Assembly Interrupt Subroutine executes a TTI instruction for each line selected every clock interrupt. The program then scans one eighth of the character assembly words to see if a full input character has been assembled for any of the lines. If a fully assembled character is found, the program stores the character and line number in the input buffer, zeros the TTI status word, and sets the TTI character assembly word to 2000. Note that bit 1 of the character assembly word is initially set to a 1 and the rest of the character assembly word is zeros. As the character is assembled, the character assembly word is shifted one bit position to the right for the start bit and each data bit. When the link can be set to a 1 by a RTR, the character is fully assembled.

7.1.2 Output Character Handling

Initially, the pseudo operation T8SOF adds start and stop bits to the output characters and places them in the second output buffer (T8OB2). Eventually, the interrupt subroutine transfers the characters from the second output buffer to the first output buffer (T8OB). One eighth of the lines are scanned for output every clock interrupt. That is, for any one line one bit may be output every eight clock interrupts. The first output buffer location for a line is tested for zero or non-zero. If it is non-zero the program outputs one bit of that location and stores the remaining information back in the first output buffer. If the T8OB location is zero, the second output buffer is tested for zero or non-zero. The second buffer location in the zero state indicates no new output. If the location is non-zero, the program outputs one bit, stores the remaining bits in the first output buffer, and zeros the second output buffer.

8. FORMAT

8.1 Input Data (TBSIR)

If the pseudo operation TBSIR skips, the input data is the following format:

8.1.1 Accumulator contains line number.

8.1.2 The lower eight bits of the register TTCHAR contain the input character.

8.3 Output Data (T8SOF)

The user presents the pseudo operation T8SOF with output characters in the following format:

8.3.1 The lower eight bits of register TTCHAR contain the output character.

8.3.2 The accumulator contains the number of the line on which the character is to be output.

9. EXECUTION TIME

9.1 Minimum

9.2 Maximum

9.3 Average

The table below indicates the percentages of machine time used for 110 baud 8-bit systems and is as accurate as is presently possible. Any additional features which may be required for the Teletype handling would add appreciably to the times shown:

TIMING TABLE
Numbers indicate the percentage of available machine time used in the average case.

<table>
<thead>
<tr>
<th>No. of Lines</th>
<th>8-Bit 110 Baud</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>34.1 %</td>
</tr>
<tr>
<td>64</td>
<td>57.7 %</td>
</tr>
<tr>
<td>96</td>
<td>81.3 %</td>
</tr>
<tr>
<td>128</td>
<td>104.9 %</td>
</tr>
</tbody>
</table>

9.4 Timing Equations

Where \( n \) = the number of lines, the 8-bit subroutines require an average time of \( 8.38n + 119.5 \) \( \mu \)sec. Clock flags (at 110 baud) occur every \( 1135 \) \( \mu \)sec.

10. PROGRAM

10.3 List of Items and Pseudo Commands

10.3.1 List of Items

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TT8BGN</td>
<td>/BEGINNING OF SUBROUTINE. MUST BE</td>
</tr>
<tr>
<td></td>
<td>/EQUATED TO START OF A PAGE. (AREA</td>
</tr>
<tr>
<td></td>
<td>/INCLUDES TWO PAGES).</td>
</tr>
<tr>
<td>TBAX1</td>
<td>/AUTOINDEX REGISTER.</td>
</tr>
<tr>
<td>TBAX2</td>
<td>/AUTOINDEX REGISTER.</td>
</tr>
<tr>
<td>TBAX3</td>
<td>/AUTOINDEX REGISTER.</td>
</tr>
<tr>
<td>TT8PG0</td>
<td>/START OF CONSTANT AREA IN PAGE 0.</td>
</tr>
<tr>
<td></td>
<td>/(LENGTH 178 REGISTERS.)</td>
</tr>
<tr>
<td>TB0BF</td>
<td>/START OF OUTPUT BUFFER. (LENGTH = n.)</td>
</tr>
<tr>
<td>TB0BF2</td>
<td>/START OF SECOND OUTPUT BUFFER.</td>
</tr>
<tr>
<td></td>
<td>/(LENGTH = n.)</td>
</tr>
<tr>
<td>TB1BF</td>
<td>/START OF INPUT BUFFER. (LENGTH = 2n.)</td>
</tr>
<tr>
<td>TB1N</td>
<td>/START OF TTI AREA. (LENGTH = 3n+1.)</td>
</tr>
<tr>
<td>TTCHAR</td>
<td>/CHARACTER AREA PAGE 0. (SINGLE REGISTER.)</td>
</tr>
</tbody>
</table>

10.3.2 List of Pseudo Command

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>Minimum</th>
<th>Average</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>TB1NIT</td>
<td>Initialize</td>
<td>-</td>
<td>-</td>
<td>42</td>
</tr>
<tr>
<td>TB8SOF</td>
<td>Skip if output free</td>
<td>24</td>
<td>-</td>
<td>42</td>
</tr>
<tr>
<td>TB8SIR</td>
<td>Skip if input ready</td>
<td>15</td>
<td>37.5</td>
<td>48</td>
</tr>
<tr>
<td>TB8SIE</td>
<td>Skip if either input ready</td>
<td>37.5</td>
<td>60.0</td>
<td>70.5</td>
</tr>
<tr>
<td>Program Listing</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>APRFT 0264</td>
<td>NUTR 0446</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ASPLG 0101</td>
<td>RETCOD 0135</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CANCE 0123</td>
<td>RPINR 0112</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DELAY 0041</td>
<td>RUPST 0006</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ENCOD 0137</td>
<td>REVEN 0651</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FIVE 1052</td>
<td>SKIPS 0052</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NPTS 0073</td>
<td>SPACE 0134</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;CLT1 0402</td>
<td>STIOP 0225</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;CLT2 0422</td>
<td>SWITCH 0690</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;CLT3 0442</td>
<td>SWIOP 0616</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;CLT4 0462</td>
<td>SWSUTR 0631</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;CLT5 0612</td>
<td>SWSKIP 0636</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AONW 0041</td>
<td>TFLT1 0412</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RRLT1 0406</td>
<td>TFLT2 0432</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RRLT2 0422</td>
<td>TFLT3 0452</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RRLT3 0442</td>
<td>TFLT4 0472</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RRLT4 0462</td>
<td>TFLT5 0622</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RRLT5 0616</td>
<td>TSLT1 0416</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RSLT1 0404</td>
<td>TSLT2 0436</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RSLT2 0424</td>
<td>TSLT3 0456</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RSLT3 0444</td>
<td>TSLT4 0476</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RSLT4 0464</td>
<td>TSLT5 0626</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RSLT5 0614</td>
<td>PFLT1 0414</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;SFLT1 0401</td>
<td>PFLT2 0434</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;SFLT2 0421</td>
<td>PFLT3 0454</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;SFLT3 0441</td>
<td>PFLT4 0474</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;SFLT4 0461</td>
<td>PFLT5 0624</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;SFLT5 0611</td>
<td>SFLT1 0411</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SP 0107</td>
<td>SFLT2 0431</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 0110</td>
<td>SFLT3 0451</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6 0111</td>
<td>SFLT4 0471</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>64 0652</td>
<td>SFLT5 0621</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FCOD 0136</td>
<td>HAITI 0241</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INFEED 0306</td>
<td>HO 0020</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>POINR 0046</td>
<td>HOIS 0027</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RACT 0125</td>
<td>HORU 1000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RACTV 0260</td>
<td>HOSTR 0045</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DFX 0050</td>
<td>HWAIT 1022</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DFX1 0051</td>
<td>HWCC 1051</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RUPR 0253</td>
<td>HWULOP 1004</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NUTALL 0200</td>
<td>NUTA 0323</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NUTCOS 0400</td>
<td>NUTIN 0400</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NUTPTS 0065</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
/BRAIT CHARACTER ASSEMBLY SUBROUTINES
/TYEP 680 TELETYPE LINE MULTIPLEXER
/LMH 7-R-A5, 8 BIT

TTRACON=3000
TTRIC=126
TBADF=600
TBIBF=420
TBDF2=500
TBX1=15
TBX2=15
TBX3=16
TCHAR=177
TBIN=337

TTL=6402 /TELETYPE INPUT COMMAND
TTO=6404 /TELETYPE OUTPUT COMMAND
TTCL=6411 /CLEAR LINE REGISTER
TTP=6414 /READ LINE REGISTER
TTSE=6412 /SET LINE REGISTER, CLR AC
TRON=6414 /TURN CLOCK ON
TTOFF=6432 /TURN CLOCK OFF
TTSK=6431 /SKIP ON CLOCK FLAG
TTIN=6431 /INCREMENT LINE REGISTER

*/TTRPG*
126 2000 TBINFL. " /INPUT READY FLAG
127 4177 TBK, THR=1 /TO RESET INPUT BUFFER POINTER
130 0000 TBL. " /-NUMBER OF LINES
131 4600 TBK7, TROHF /K FOR 1ST OUTPUT BUFFER
133 3021 TBK8, TROCM /TO ENTER COMMON ROUTINE
133 3151 TBKOUT, TROUTS /SKIP IF OUTPUT FREE
134 3200 TAMS, TAIN /SKIP IF INPUT READY
135 3224 TANS, TACM /INITIALIZE ROUTINE
136 3344 TRS, TASE /SKIP IF EITHER LINE READY
137 4600 TBURTK, TROHF /POINTER TO 1ST OUTPUT BUFFER
140 5000 TBURTK2, TROHF2 /POINTER TO 2ND OUTPUT BUFFER
141 5000 TBURTK3, TROHF3 /K FOR 2ND OUTPUT BUFFER
142 7777 TBON1, -16 /HOLD MAJOR LOOP COUNTER
143 9000 TBON2, " /MINOR LOOP COUNTER
144 9000 TBON3, " /COUNTER FOR INPUT BUFFER

*/TTRPG*
/MULTIPLE LEVEL INTERRUPT ROUTINE
/ALLOWS MULTIPLE LEVEL INTERRUPT TO THIS ROUTINE AND UNLIMITED OTHERS

*/TADIS/ 157 TBLC /LEVEL COUNTER
000 2334
001 5216 JMP TADIS3 /2ND LEVEL INTERRUPT
002 3335 DCA TASA /SAVE ACCUMULATOR
003 7010 RAR /GET LINK
004 3336 DCA TRSVK /SAVE LINK
005 1900 TAN 2 0 /INTERUPT ADDRESS
006 3337 DCA TRSVA /SAVE ADDRESS
007 6414 TRL /RN LINE NUMBER
008 3340 DCA TRSVN /SAVE LINE NUMBER
009 8434 TTRON /CLEAR FLAG ONLY
010 4001 TBOP1 /H-F-ENABLE PROGRAM INTERRUPT
011 3350 TAN T8K9 /STARTING LINE-1
%014 6413 TTSL+1 /SET LINE REGISTER, CLEAR AC
%015 5741 JMP T8K1 /JUMP TO TTI LOOP

/2ND LVFL INTERRUPT

%016 6434 T80153, TRON /CLEAR CLOCK FLAG
%017 6901 ION /RE-FNABLE PROGRAM INTERRUPT
%020 5480 JMP I Z P /RETURN TO THE MAIN PROGRAM

/RETURN FROM INPUT TTI LOOP

%021 1342 T8COM, TAN T8MNC /MINOR COUNTER, NUMBER OF LINES/A
%022 3143 DCA 7 T8CNT2 /MINOR LOOP COUNTER
%023 1343 TAN TALN /LINF NUMBER
%024 6413 TTSL+1 /SET LINE NUMBER
%025 1537 T8COMA, TAN I Z T8OUTK /OUTPUT WORD
%026 7450 SNA /SOMETHING TO TRANSMIT
%027 5333 JMP T8COM8 /SEE IF WORD AVAILABLE
%028 6405 TNO+1 /MINOR LINE REGISTER AND OUTPUT
%029 3737 DCA 1 Z T8OUTK /STORE WORD
%032 1414 T8COM1, TAN I Z T8X1 /PICK UP CHARACTER ASCII WORD
%033 7110 CLL RAR /PUT RIT 11 IN LINK
%034 7430 S2L /CHARACTER NOT COMPLETED
%035 5301 JMP T8COM6 /STORE CHARACTER
%036 7200 CLA /CLEAR AC FOR TAN
%037 2137 T8COM3, ISZ 2 T8OUTK /UPDATE OUTPUT ADDRESS
%040 2140 ISZ 2 T8OK2 /UPDATE 2ND BUFFER ADDRESS
%041 2014 ISZ 2 T8X1 /UPDATE FOR NEXT INPUT LINE
%042 2014 ISZ 2 T8X1 /UPDATE FOR NEXT INPUT LINE
%043 2143 ISZ 2 T8CNT2 /ARE ONE-EIGHTH OF LINES CHECKED?
%044 6225 JMP T8COM9 /CHECK NEXT LINE

%045 6414 T8COM4, TTRL /READ LINE NUMBER
%046 3343 DCA T8LN /SAVE LINE NUMBER
%047 2142 ISZ 2 T8CNT1 /HAVE ALL LINES BEEN CHECKED
%049 5263 JMP T8COM5 /RESET AND DISEMISS
%051 1344 TAN T8K2 /-10
%052 3142 DCA 2 T8CNT1 /RESET MAJOR LOOP COUNTER
%053 1345 TAN T8K3 /TRIM+1
%054 3814 DCA 2 T8X1 /RESET INPUT LINE POINTER
%055 1350 TAN T8K3 /STARTING LINE-1
%056 3343 DCA T8LN /RESET LINE NUMBER
%057 1171 TAN 2 T8K7 /T0RF
%058 3137 DCA 2 T8OUTK /RESET OUTPUT LINE POINTER
%059 1141 TAN 2 T8K3A /T0RF2
%060 3140 DCA 2 T8OUTK /RESET 2ND BUFFER POINTER
%063 6002 T8COM5, TOF /TURN OFF INTERRUPT
%064 7240 STA /-1
%065 1334 TAN T8LC /LEVEL COUNTER
%066 3334 DCA T8LC /RESTORE LEVEL COUNTER
%067 1334 TAN T8LC /LEVEL COUNTER
%067 7700 SMA CLA /RESTORE AC ETC
%069 5212 JMP T80152 /CHECK INPUT AGAIN, ETC.
%072 1340 TAN TBSVLN /LINE NUMBER
%073 6413 TTSL+1 /SET LINE REGISTER, CLEAR AC
%074 1336 TAN TBSVLK /PICK UP LINK
%075 7104 CLL PAL /RESTORE LINK
%076 1335 TAN TBSA /RESTORE ACCUMULATOR
%077 6901 IOM /EMARLF INTERRUPT
3100 5737  JMP 1 TB5VM  /RETURN TO MAIN PROGRAM
3101 7112  TBCOM6.  CLI  FTR  /REMOV START CODE
3102 4415  P50A 1, 2  TRAX2  /STORE CHARACTER
3103 4414  TRPL  /RENAME LINE NUMBER
3104 4415  P50A 1, 2  TRAX2  /STORE LINE NUMBER
3105 9114  TAN 2  TRAX1  /PICK UP ADDRESS POINTER
3106 4346  TAN  TRAX5  /-
3107 3014  P50A 2  TRAX1  /RESET POINTER
3108 4414  P50A 1, 2  TRAX1  /WORD STATUS AND COUNTER WORD
3109 1137  TAN  TRAX6  /WORD TO RESTORE ASSEMBLY WORD
3110 4414  P50A 1, 2  TRAX1  /RESET CHARACTER ASSEMBLY WORD
3111 2126  ISZ  T8INF  /SET INPUT READY FLAG
3112 7144  ISZ  T8CNT3  /HAS END OF BUFFER BEEN REACHED
3113 6237  JMP  TBCOM3 /UPDATE REGISTERS
3114 1127  TBCOM7,  TAN 2  TBPKF  /TRHF-1
3115 1015  P50A 2  TRAX2  /RESET INPUT BUFFER ADDRESS
3116 1130  TAN  TANL  /LENGTH OF BUFFER, NUMBER OF LINES
3117 3144  P50A 1, 2  TRAX2  /RESET LENGTH COUNTER
3118 5232  JMP  TBCOM3 /UPDATE REGISTERS
3119 1540  TBCOMA,  TAN 2  T8OTK2  /PICK UP DOUBLE-RUFFRED WORD
3120 7440  S2A  /NOTHING TO SEND
3121 5330  JMP  .+3  /SFAND NEW WORD
3122 4401  T8INCR  /INCREMENT LINE REGISTER
3123 5232  JMP  TBCOM1 /CONTINUE FOR INPUT
3124 1545  T8O+1,  T8+1  /INCR. LINE REGISTER AND OUTPUT
3125 3537  P50A 1, 2  T8OUTK  /STORE WORD
3126 3342  P50A 12  T8OTK2  /ZERO 240 WORD
3127 6232  JMP  TBCOM1 /CONTINUE FOR INPUT

/CONSTANTS
3134 7777  T8LC,  -1  /INTERRUPT LEVEL COUNTER
3135 9000  T8SA,  0  /SAVE ACCUMULATOR
3136 9000  T8SLK,  0  /SAVE LINK
3137 9000  T8SVK,  0  /SAVE PROGRAM COUNTER
3138 9000  T8SVN,  0  /SAVE LINE NUMBER
3139 3377  T8K1,  T8IN  /START OF TTI SERIES
3140 3377  T8K1,  T8IN  /MINOR LOOP COUNTER. NO OF LINES/8
3141 3377  T8K1,  T8IN  /LINE NUMBER
3142 7770  T8K2,  -1M  /TO RESET MAJOR LOOP COUNTER
3143 3400  T8K3,  T8IN+1  /TO RESET INPUT LINE POINTER
3144 7776  T8K5,  -2  /FOR SUBTRACTION
3145 2000  T8K6,  2000  /TO RESET 8-RIT ASSEMBLY WORD
3146 0000  T8K9,  0  /STARTING LINE-1

/SEED OPERATIONS
3147 0000  T8OF=JMP 1 7 T8OUTS  /STARTING LINE-1
3148 0000  T8OUTS,  0  /SEED OUTPUT IS FREE AND TRANSMIT CHARACTER AT T8CHAR
3149 0000  T8OUTS,  0  /OTHERWISE DONT SKIP
3150 0000  T8OUTS,  0  /LINE NUMBER MUST BE IN AC. 24US MIN. 42US MAX.
3151 0000  T8SOF=JMP 1 7 TASOUTT
3152 0370  AND T8K12  /177
3153 1371  TAN  T8SL  /STARTING LINE NO.
3154 1141  TAN 2  T8K36  /OUTPUT BUFFER ADDR
3155 3372  NCA  T8WA  /WORK AREA
3156 1772  TAN 1  T8WA  /OUTPUT CHARACTER
3157 7640  S2A CLA  /SKIP IF FREE
3158 5751  JMP 1 T8OUTs  /EXIT
3161 1177 TAD 2 TCHWTCH /PICK UP CHARACTER
3162 1177 AND TAK11 /8 BITS ONLY
3163 1174 TAD TAK12 /1400 FOR STOP CODE
3164 7194 CALL PAL /CREATE START CODE
3165 3772 0CA 1 TAW4 /STORE CHARACTER IN TABLE
3166 3771 052 TAW4 /INDEX EXIT
3167 7191 JMP 1 TOUTS /EXIT
3168 0177 TAK10, 177 /FOR LINE NUMBER
3169 0000 TBLI, 0 /STARTING LINE NUMBER
316a 0000 TBAI, 0 /WORK AREA
316b 0377 TAK11, 377 /FOR EIGHT BIT CODE
316c 0175 TAK12, 1400 /FOR STOP CODE
0177 0177 TAK10, 177 /FOR CHARACTERS AVAILABLE AND RETURN WITH LINE NO. IN AC
0178 TAD TCHWTCH /CHAR AT TCHWG
0179 /OTHERWISE DO NOT SKIP. 15 US MIN, 48 US MAX, 37.5 US NORMAL
017a IF READY
017b TDOIN=JMS 1 2 TAIN
017c 0172 TAIN, 0
017d 6002 10F
017e 7240 CLA CMA /SET AC TO -1 FOR TAD
017f 1126 TAD 2 TAILF /INPUT FLAG COUNTER
017g 7510 SPA /SOMETHING AVAILABLE
017h 5221 JMP TAIN4 /EXIT
017i 3726 NCA 2 TAILF /RESTORE FLAG COUNTER
017j 2223 ISZ TAIN4 /END OF BUFFER? STARTS AT -N-1
017k 6215 JMP, +5 /GET CHARACT
017l 1130 TAD 2 TAIL /-NUMBER OF LINES
017m 2223 NCA TAIN4 /RESET COUNTFR
017n 1127 TAD 2 TAWK /BUFFER ADDRESS-1
017o 8016 0CA 2 TACK /RESET ADDRESS
017p 1416 TAD 1 2 TAWX /PICK UP CHARACTER
017q 1717 0CA 2 TCHWTCH /STORE CHARACTER
017r 1416 TAD 1 2 TAUX3 /PICK UP LINE NUMBER
017s 2220 ISZ TAINS /INDEX EXIT
017t 6201 TAIN4, 10m
017u 6200 JMP TAINS /EXIT
017v 2208 TBLI, 0 /-NO OF LINES
017w 0000 TAIN4, 0 /INITIALIZATION
017x /ENTER WITH NUMBER OF LINES IN AC
017y /FORMAT TAIN 1 2 TAIN 1 2 TAIN /1ST LINE NO.
017z TAINI=JMS 1 2 TAIN
0180 0000 TBOOS, 0
0181 8291 AND TAK14 /377
0182 1130 0CA 2 TAIL /STORE NUMBER OF LINES
0183 1130 TAD 2 TAIL /NUMBER OF LINES
0184 0322 AND TAK15 /7
0185 7640 SZA CLA /MULTIPLE OF 8?
0186 1232 TAD TAK16 /10
0187 1130 TAD 2 TAIL /NUMBER OF LINES
0188 9324 AND TAK17 /370
0189 7041 C1A /-TWO'S COMP NUMBER OF LINES
018a 1130 0CA 2 TAIL /-N, CONSTANT
018b 1130 TAD 2 TAIL /-N
018c 3144 0CA 2 TAIN3 /INPUT COUNTER

Digital-8-35-5-8
Page 11
TAN TRK20 /THIN-1
DCA 2 TRAX1 /TO STORE TTI TARLF
TAN TRK21 /TARLF-1
DCA 2 TRAX2 /TO CLEAR OUTPUT AREA
TAN TRK37 /TBORF-1
DCA 2 TRAX3 /TO CLEAR DOUBLE BUFFER
TAN 2 TRML /-1
DCA TRCNT4 /FOR COUNTING
DCA TRGOS, TAN TRK22 /TTI+INCR
DCA 1 2 TRAX1 /STORE TTI
DCA 1 2 TRAX1 /ZERO STATUS AND COUNTER WORD
TAN TRK23 /ASSEMBLY RESET WORD
DCA 1 2 TRAX1 /RFSET ASSEMBLY WORD
DCA 1 2 TRAX2 /ZERO OUTPUT WORD
DCA 1 2 TRAX3 /CLEAR DOUBLE BUFFER
ISP TRCNT4 /COUNTER
JMP TRGOS /DON NEXT LINE
TAN TRK24 /JMP I 2 TRKB
DCA 1 2 TRAX1 /STORE FINAL JUMP
DCA 1 TRAX1 /-1
RTO /DIVIDE BY 4
ROR /DIVIDE BY 8
AND TRK25 /17
DCA I TRK26 /7760, MAKE NUMBER NEGATIVE
DCA I TRK27 /TMNC
STA /-1
DCA TRCNT4 /SET COUNTER TO SKIP 1ST TIME
JMP TRFFX /TRIRF-1
DCA 2 TRAX2 /SET INPUT BUFFER POINTER
TAN TRK2R /-10
DCA 2 TRAX2 /MAJOR LOOP COUNTER
DCA 2 TRCNT1 /TRIN+1
TAN TRK37 /TAN TRK22 /TRK9, STARTING LINE NO. -1
DCA 1 TRAX1 /STOR TTI POINTER
TAN 2 TRK7 /TARLF
DCA 2 TRAX1 /1ST OUTPUT BUFFER POINTER
TAN 2 TRK34 /TARLF
DCA 2 TRAX2 /2ND OUTPUT BUFFER POINTER
STA /-1
TAN 1 TRGOS /STARTING LINE NO.
DCA 1 TRK33 /TARLF, STARTING LINE NO. -1
DCA 1 TRK33 /TARLF
DCA 1 TRAX1 /MAKE NEGATIVE
DCA 1 TRAX4 /TARLF - STARTING LINE NO.
DCA 2 TRAX1 /CLEAR INPUT FLAG COUNTER
STA /-1
DCA 1 TRK35 /TARLF, RFSET INTERRUPT LEVEL COUNTER
DCA 1 TRAX1 /INDEX EXIT
JMP 1 TRGOS /EXIT

/CONSTANTS
T8K14, 377 /FOR LINE NUMBER
T8K15, 377 /FOR EVEN MULTIPLE OF 8
T8K16, 19 /FOR EVEN MULTIPLE OF 8
T8K17, 377 /FOR EVEN MULTIPLE OF 8
T8K18, 1 /FOR COMPLEMENTING
T8K20, TAIN-1 /FOR STORING TTI'S
T8K21, TBORF-1 /FOR OUTPUT AREA
3330 2403  T8K22,  TTI+1  /TI + INCREMENT
3331 2000  T8K23,  2IN0  /ASSEMBLY RESET WORD
3332 5532  T8K24,  JMP 1 7  T8K8  /FOR FINAL JUMP
3333 3017  T8K25,  17  /FOR -N/A
3334 7760  T8K26,  7760  /FOR MAKING NEGATIVE
3335 3142  T8K27,  T8MNC  /FOR -N/A
3336 7770  T8K28,  18  /FOR MAJOR LOOP COUNTER
3337 3400  T8K30,  T8IN+1  /FOR TTI POINTER
3340 3150  T8K33,  T8K9  /FOR STARTING LINE-1
3341 3171  T8K34,  T8SL  /-STARTING LINE NO.
3342 3134  T8K35,  T8LC  /FOR INTERRUPT LEVEL COUNTER
3343 4777  T8K37,  T8DF2-1  /FOR DOUBLE BUFFER
/SKIP IF CHARACTER AVAILABLE FROM EITHER OF TWO TYPES OF LINES
/OTHERWISE DO NOT SKIP.  31.5US MIN, 74.5US MAX, 63US NORMAL
/IF READY
T8SIF=JMS 1 7  T8IES
3344 0000  T8SE,  1
3345 4534  T8SIV  /CHECK 8-BIT CODE
3346 5381  JMP 1 7  /CHECK 5-BIT CODE
3347 2344  T8S TASE  /INDEX EXIT
3348 1744  JMP 1  T8SE  /EXIT
3349 1377  T8S1W  /CHECK 5-BIT CODE
3350 2574  JMP 1  T8SE  /EXIT
3351 2344  T8S TASE  /INDEX EXIT
3352 5744  JMP 1  T8SE  /EXIT
PAUSE
11. DIAGRAMS

11.1 Flow Charts

INITIALIZATION

(AC) • NUMBER OF LINES ENTRY

1. TBGS

CLR AC TO LAST 8 BITS

STORE AC IN TBNL

NO

OF LINE
AN EVEN MULT
OF 8?

YES

NO

LOAD B INTO AC

CLEAR AC

(AC) + (TBNL) CLR AC BITS 9, 10, 11

2'S COMPL (AC) STORE IN TBNL

TBNL+2'S COMPL OF THE NO. OF LINES TO THE NEXT MULTIPLE OF 8

TBNL TO TBCNT3

TBCNT3 • INPUT COUNTER

LOAD INDEX REGISTERS

AX1 • TTI TABLE

AX2 • OUTPUT AREA

AX3 • DOUBLE BUFFER

CNT4 • NO. OF LINES

LOAD TBG02

LOAD AC WITH TTIincr AND TTI instr

DCA AX1

DCA AX2

SET INCR AND TTI INTO TTI TABLE

LOAD AC WITH 2000

DCA AX1

DCA AX2

DCA AX3

SET TTI + 2

CLR BUFFERS

LOAD EXIT JUMP INTO TTI TABLE

GOT WHOLE TABLE ISZ TBCNT4?

YES

NO
INITIALIZATION

1. \( \text{AC} = \text{NUMBER OF LINES} \)
2. \( \text{DIVIDE NUMBER OF LINES BY 8} \)
3. \( \text{MAKE NEGATIVE STORE IN TMNC} \)
4. \( \text{SET TBCNT4 TO ALL 1'S} \)
5. \( \text{SET INPUT BUFFER POINTER} \)
6. \( \text{SET TBCTNT1 TO -B} \)
7. \( \text{TBCNT1 MAJOR LOOP COUNTER = -B} \)
8. \( \text{SET TT1 POINTER} \)
9. \( \text{SET OUTPUT BUFFER POINTERS} \)
10. \( \text{STARTING LINE NO.-1} = \text{STARTING LINE} \)
11. \( \text{COMPLEMENT STARTING LINE NO.1} \)
12. \( \text{RESET INTERRUPT LEVEL COUNTER} \)
13. \( \text{EXIT IN IT} \)

OUTPUT

1. \( \text{ENTER TBOUTS} \)
2. \( \text{CLEAR AC TO LINE NUMBER} \)
3. \( \text{SUBTRACT STARTING LINE NUMBER} \)
4. \( \text{ADD OUTPUT BUFFER STARTING ADDRESS} \)
5. \( \text{LOAD AC WITH CONTENTS OF 2ND BUFFER} \)
6. \( \text{OUTPUT FREE?} \)
   - NO \( \rightarrow \text{EXIT} \)
   - YES \( \rightarrow \text{OUTPUT CHARACTER} \)
7. \( \text{INSERT START AND STOP CODES} \)
8. \( \text{STORE IN OUTPUT BUFFER} \)
9. \( \text{INCREMENT EXIT FOR SKIP} \)
**2ND LEVEL**
(TBCNT2)=#LINES/B

**SAVE AC** 3.0
**SAVE LNK** 4.5
**SAVE I ADDRESS** 6.0
**SAVE LINE #** 4.5
**CLOCK ON** 1.5

**19.5μ sec**

**ION** 1.5
**SET LINE** 4.5
**JMP TBIN** 3.0

**3**

**ZEROS SECOND BUFFER**
**ADDS 4.5**

**5**
**3.0**
**3.0**
**6.0**

**SET LINE REGISTER** (TBLNI)

**5**

**SET MINOR LOOP COUNTER** (TBCNT2)=#LINES/B

**6.0**

**TBCOM**

**6.0**

**TBCOMB**

**3**

**increment line register**

**4.5**

**Save rest of character**

**34.5 sec** NO INPUT/PARTIAL OUTPUT LEFT

**43.0 sec** NO INPUT/NO NEW OUTPUT

**47.5 sec** NO INPUT/NEW OUTPUT CHARACTER

ADD 39.5 FOR EACH COMPLETE INPUT
REFERENCES (Not Applicable)