TRANSCODE

A SYSTEM OF CODING FOR THE PERSHATTI MAP I COMPUTER

BY

B. H. WORSLEY
J. N. F. HUME

CANADA
UNIVERSITY OF TORONTO
COMPUTATION CENTRE
In the use of a flow-chart-point system of notation.

In many cases the speed of the machine code is only that important

the main loss of speed in working with TRANSCODE

presents unusual difficulties. It is beneficial that

Every work in production processes for what solutions

in dynamic single-shot programs, and in dynamic analyses

TRANSCODE is expected to be practically useful

for learning to operate. In the TRANSCODE manner.

However, a study of this appendix is not necessary

and the Tranmer system of input and organization

procedures remain with the permanent mechanical code

in appendix is also introduced for the benefit of

described, assuming such a framework.

imply that the system will be the computer. The system must therefore be one of the possibilities of the system which makes it possible to learn and apply

TRANSCODE is intended to provide a system of

end input of both data and instructions

read for programming organization of the program

in probabilistic-decimal manner. The programs are pro-

three-addresses, mnemonic code of instructions operating

which enables a personal work in computer to utilize a

TRANSCODE is a comprehensive system of routines

PRoPRIETARY ROUTINE

DATE 1 Oct., 1964

NAME TRANSCODE

CLASS Composite

PRoPRIETARY ROUTINE
Deviations in operational instructions must not be listed as the result of
being modified, and may not be written in operational instructions. A de
revised operational instruction may be referred to as a revised operational in
struction. When operational instructions are revised, the code of the opera-
tional instruction must be broken down into a logical sequence of operations. 

The code of operation instructions is given on page 6 in the General Informa-
tion, and the code of instruction is given on page 7.

Transferred to and from the drum

The program counter is an intermediate variable used to indicate the

The main diagonal of the electronic score, the Z sequence, can change
on a page in the electronic score. The drum is programmed by the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the

The program counter is an intermediate variable used to indicate the
The input and organization of the execution of the program, however, is not necessary to explain or describe further. Any explanation of the execution of the program can be postponed until later.

TRANSCODE

STOP. These functions were called into a context only when
seen and read so that they are not necessary to present a

stop (eight) is provided at the end of

As means of PREP read as an operational limitation.

STOP. These functions were called into a context only when
seen and read so that they are not necessary to present a

stop (eight) is provided at the end of

As means of PREP read as an operational limitation.
NO effect. Can proceed with tape control.

The sequence of instructions into a form suitable for
insertion into the output of all transcribed
instructions into a form suitable for
insertion into the output of all transcribed

STOP

Drum positions are displayed when the scanner reads
a drum position, and the

DUMM COM

To read in at a page of at most 21 numbers

To read in the set of counters. Assistant

QUIT

number of decimal points, and the point read by
instructions printed with instructions
To read in the program, INS 000 should

TRANSCODER

-4 -
\[ y = x^2 + 2x + 1 \]

Place the function numerator \( f \) and denominator \( g \) in \( \frac{[y]}{[2]} \).

Operational Instructions:

1. Initialize variables
   - \( x = 0 \), \( y = 0 \), \( b = 0 \), \( p = 0 \)

2. Increment \( b \)

3. For each \( x \) from 0 to \( b-1 \):
   - Increment \( y \)
   - Print \( x, y, b, p \)

4. Exit loop when \( y \) is greater than \( b \)

5. End program

Note: This is a text-based representation of the operational instructions. Actual implementation may require additional context or code.
Electron scope.

Note

In terms of the actual meaning, when an instruction is

<table>
<thead>
<tr>
<th>0.000</th>
<th>0.000</th>
<th>0.000</th>
<th>0.000</th>
<th>0.000</th>
<th>0.000</th>
<th>0.000</th>
<th>0.000</th>
<th>0.000</th>
<th>0.000</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRANS</td>
<td>INDOR</td>
<td>WHIP</td>
<td>T0.000</td>
<td>X0.000</td>
<td>Y0.000</td>
<td>X0.000</td>
<td>Y0.000</td>
<td>X0.000</td>
<td>Y0.000</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Follow up:

The following instructions should be immediately into a loop as

IF T is to be taken on the values O', T', T' -- T, T' in succession.

<table>
<thead>
<tr>
<th>0.000</th>
<th>0.000</th>
<th>0.000</th>
<th>0.000</th>
<th>0.000</th>
<th>0.000</th>
<th>0.000</th>
<th>0.000</th>
<th>0.000</th>
<th>0.000</th>
</tr>
</thead>
<tbody>
<tr>
<td>WHITE</td>
<td>O01.0</td>
<td>B01.4</td>
<td>P00.0</td>
<td>O00.0</td>
<td>P00.0</td>
<td>O00.0</td>
<td>P00.0</td>
<td>O00.0</td>
<td>P00.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For example, write the X page on drum position

white x page on drum position

By direct inspection and modification of some b-value

Also be modified processes, but this must be done

Instructions referring to drum storage location can

<table>
<thead>
<tr>
<th>0.000</th>
<th>0.000</th>
<th>0.000</th>
<th>0.000</th>
<th>0.000</th>
<th>0.000</th>
<th>0.000</th>
<th>0.000</th>
<th>0.000</th>
<th>0.000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z000</td>
<td>X0.0</td>
<td>T0.0</td>
<td>Z000</td>
<td>X0.0</td>
<td>T0.0</td>
<td>Z000</td>
<td>X0.0</td>
<td>T0.0</td>
<td>Z000</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The following Z0:

These 2 instructions can therefore be used in place of

<table>
<thead>
<tr>
<th>0.000</th>
<th>0.000</th>
<th>0.000</th>
<th>0.000</th>
<th>0.000</th>
<th>0.000</th>
<th>0.000</th>
<th>0.000</th>
<th>0.000</th>
<th>0.000</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRANS</td>
<td>Z000</td>
<td>X0.0</td>
<td>TRANS</td>
<td>Z000</td>
<td>X0.0</td>
<td>TRANS</td>
<td>Z000</td>
<td>X0.0</td>
<td>TRANS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For example: Place zero in each of the instructions X0.0.

X0.0 -- X0.0, beginning with X0.0.

Note: Any other process should be used in conjunction with the above instructions to

In order to determine the total number of instructions...
Production of all procedures must be completed, then printed.

\[ \frac{1}{2} \times \text{normalized number} \geq \frac{1}{4} \times 1 \text{ sec.} \]

The result is the program output, a snap accuracy (SP/9) DS/O. If the result is too large, use a dynamic snap accuracy. If the result is too small, the program output is fine.

If the normalized number is too large, output operations are continued. If the normalized number is too small, output operations are stopped.

No clue is given at about 3 sec. for exponents of order 1000, the delay is about 2 sec. for exponents of order 10. When a result is a floating point in the conversion of numbers, there is a slight delay in the conversion of numbers.

<table>
<thead>
<tr>
<th>p</th>
<th>2p</th>
<th>5p</th>
<th>10p</th>
<th>15p</th>
<th>20p</th>
<th>25p</th>
</tr>
</thead>
<tbody>
<tr>
<td>100 m sec.</td>
<td>COKE</td>
<td>50 m sec.</td>
<td>ZENO</td>
<td>70 m sec.</td>
<td>COKE</td>
<td>TINS</td>
</tr>
<tr>
<td>5 m sec.</td>
<td>WHIRL</td>
<td>80 m sec.</td>
<td>REAP</td>
<td>150 m sec.</td>
<td>1/2 GPR</td>
<td>DROWN</td>
</tr>
<tr>
<td>All-time instructions</td>
<td>165 to 200 m sec.</td>
<td>60 m sec.</td>
<td>165 to 210 m sec.</td>
<td>115 to 200 m sec.</td>
<td>120 to 200 m sec.</td>
<td>70 to 90 m sec.</td>
</tr>
</tbody>
</table>

Transaction into read code, 4 or 5 instructions per sec.

Reading-in of NMB or ONSL, 25 m sec. per decimal digit.

Reading-in of INS1, 2 or 3 instructions per sec.

*All programs are run on the test program.

Print-checking and re-performing the test program.

The test program and the production program are the same. To produce instructions in the production program, the print-checking must be identical to the test program. If the test program is the same, the computer, it may be found useful to print our internal.

In the development stages of testing a program can

TRANSCODE

PRINT-CHECKING
The symbols $\neq$ and $\not=$ can be used to indicate "not equal." The word "the" is printed on the page in the form "THE.

"The" word end of the page in the FANAC reader pattern receiver is shown to read "ANTER.

The punch end of any complete FANAC rod is punched "J" style and of any complete FANAC tape is punched ".J" style.

The tape counter shows whether the read head has reached the end of the tape and terminated.

To read for sequence, the tape should be punched once on each keypunch and each.

Hand-punching the master tape:

- For V4PD, A4A and all so on, both number and tape need to be punched.
- For V4PD, A4A, B4B, and all so on, both number and tape need to be punched.
- For V4PD, A4A, B4B, and all so on, both number and tape need to be punched.

The number should be punched as 1+100.

The number zero must be accurately defined in loading.

The numbers of main control and the numbers of control words must be punched and defined, respectively.

The number of main control and the numbers of control words must be punched and defined, respectively.

The number of main control and the numbers of control words must be punched and defined, respectively.

Any number of non-sufficient digits are acceptable.

To read in the preparation of numerical tape, too

Meaning of digits:

- Digits may be punched, but only the first 10 will be read. Digits need to be punched. Any digit must be punched only the first 10, otherwise the first 10 will be punched.

If $n = 0$, it is sufficient to punch "B".

If $n > 0$, it is sufficient to punch "A".

The generated position decimal number is punched as

---

FANAC CONTROL
The tape to store this program should be punched as follows:

<table>
<thead>
<tr>
<th>0000</th>
<th>0000</th>
<th>0000</th>
<th>0000</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>0000</td>
<td>0000</td>
<td>0000</td>
</tr>
<tr>
<td>002Z</td>
<td>0010</td>
<td>0010</td>
<td>0010</td>
</tr>
<tr>
<td>0020</td>
<td>0200</td>
<td>0200</td>
<td>0200</td>
</tr>
<tr>
<td>0020</td>
<td>0000</td>
<td>0200</td>
<td>0200</td>
</tr>
<tr>
<td>0000</td>
<td>9000</td>
<td>0000</td>
<td>0000</td>
</tr>
<tr>
<td>0000</td>
<td>9000</td>
<td>0000</td>
<td>0000</td>
</tr>
<tr>
<td>0020</td>
<td>0200</td>
<td>0000</td>
<td>0000</td>
</tr>
<tr>
<td>0020</td>
<td>0000</td>
<td>0200</td>
<td>0200</td>
</tr>
</tbody>
</table>

The operations to be performed are as follows:

\[ \begin{align*}
& \text{CON} = 0 \\
& \text{SPE} = 0 \\
& \text{FEN} = 0 \\
& \text{FHN} = 0 \\
& \text{NUM} = 0 \\
& \text{DRM} = 01 \\
& \text{CON} = 0 \\
& \text{CON} = 0 \\
& \text{SPE} = 0 \\
& \text{SPE} = 0 \\
\end{align*} \]
\[
\text{let } \sum_{\text{and } z=0}^n \text{ contain } (n + 1) \text{ the } \frac{1}{t}\left(\frac{i+u}{l+t+u}\right) \\
\text{exhibit, } Z_{\text{contains}} \text{ used to test } Z_{\text{is needed to retain }} Z_{\text{is any}}
\]

The method used above is quite direct. Answers are

<table>
<thead>
<tr>
<th>ENTR</th>
<th>STOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cnst</td>
<td>I + 1</td>
</tr>
</tbody>
</table>

| - | - | - | 011 |
| - | - | 005.0 | 010 |
| 0.0 | 0.0 | 005.4 | 009 |
| 0.0 | 0.0 | 000.6 | 009 |
| 0.0 | 0.0 | 020.0 | 020 |
| 0.0 | 0.0 | 020.5 | 020 |
| 0.0 | 0.0 | 000.5 | 000 |
| 0.0 | 0.0 | 020.0 | 020 |
| 0.0 | 0.0 | 020.0 | 020 |
| 0.0 | 0.0 | 020.0 | 020 |
| 0.0 | 0.0 | 020.0 | 020 |

A program to do this could be written as follows:

\[
\text{FOR } n = 1, 2, \ldots \text{ to five significant}
\]

\[
\text{decimal places, } i \text{ must be } \text{and in blocks of}
\]

\[
\text{transcode}
\]

11 -
The program should be open. If not, press the escape key. If the program is already open, press the enter key. If the program is not running, press the enter key.

Before the program is started, the user should be informed of the necessary precautions and the possible consequences of not following them. The program should be started by pressing the enter key. If the program is not running, press the enter key. If the program is already open, press the enter key.

After the program is started, the user should be informed of the necessary precautions and the possible consequences of not following them. The program should be started by pressing the enter key. If the program is not running, press the enter key. If the program is already open, press the enter key.

The program should be open. If not, press the escape key. If the program is already open, press the enter key. If the program is not running, press the enter key.

Before the program is started, the user should be informed of the necessary precautions and the possible consequences of not following them. The program should be started by pressing the enter key. If the program is not running, press the enter key. If the program is already open, press the enter key.

After the program is started, the user should be informed of the necessary precautions and the possible consequences of not following them. The program should be started by pressing the enter key. If the program is not running, press the enter key. If the program is already open, press the enter key.

The program should be open. If not, press the escape key. If the program is already open, press the enter key. If the program is not running, press the enter key.

Before the program is started, the user should be informed of the necessary precautions and the possible consequences of not following them. The program should be started by pressing the enter key. If the program is not running, press the enter key. If the program is already open, press the enter key.

After the program is started, the user should be informed of the necessary precautions and the possible consequences of not following them. The program should be started by pressing the enter key. If the program is not running, press the enter key. If the program is already open, press the enter key.
order in the program, just translated.

NSP to be preceded immediately after the last meaningful
right instruction to be executed for the machine order.

Read 001, 002, etc. and execute on a separate set
consecutive drum locations normally used for the

2.

of these instructions. In the event of programs to be
implemented as NSP's, instead of the preceding

2. to be preceded in the order of

TRANSCODES.
\[ \text{For } n = 3 \text{, } g \neq 6, \text{ } g + \text{ } \overline{g} \leq 100,1750. \]

\[ \text{Integer } \leq 100,1750. \]

or, approximately, \( g \) is any integer satisfying

\[ \left( 1 + \log_{10} n \right) \text{ of } 1000. \]

In general,

\[ \left( \frac{1}{\text{Unit Digit}1000} \right) \text{ of } 1000. \]

NOTES: Only even integers may be input in this manner.

\begin{align*}
+4+1 & \ 00 \\
+2+1 & \ 06 \\
+11+3 & \ 08 \\
+6+6 & \ 10 \\
+6+3 & \ 16 \\
+6+6 & \ 20 \\
+6+6 & \ 24 \\
+6+6 & \ 28 \\
+6+7 & \ 32 \\
+1+1 & \ 34 \\
+1+2 & \ 36 \\
+2+2 & \ 38 \\
+2+3 & \ 40 \\
+2+3 & \ 42 \\
+2+3 & \ 44 \\
+2+3 & \ 46 \\
+2+1 & \ 48 \\
+8 & \ 08 \\
+4 & \ 06 \\
+2 & \ 04 \\
+1 & \ 02 \\
1 & \ 00 \\
0 & \ 00 \\
\end{align*}

To input the integer into the exponent zone of a TRANCODE.

In general, punch as follows, with numeric

\[ \text{INTEGRAL} \]

\[ \text{TRANSCODE} \]

\[ -10- \]

\[ + \]