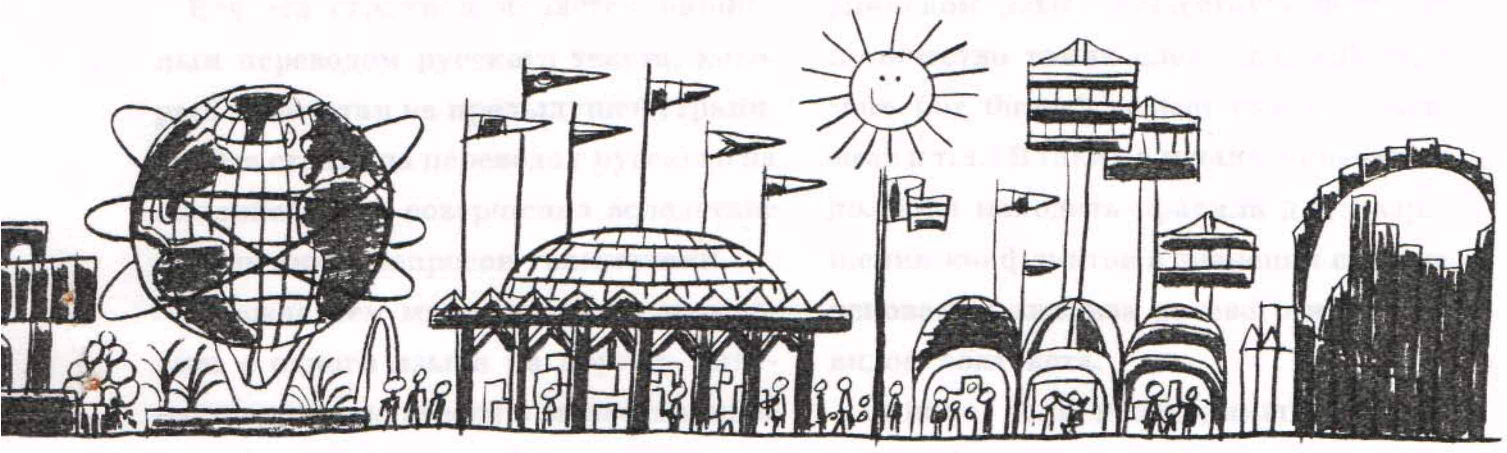


THE WORLD'S FAIR MACHINE TRANSLATOR

Dan M. Bowers and Miles B. Fisk,
Research Staff Members, T. J. Watson Research Center, IBM Corp., Yorktown Heights, N.Y.

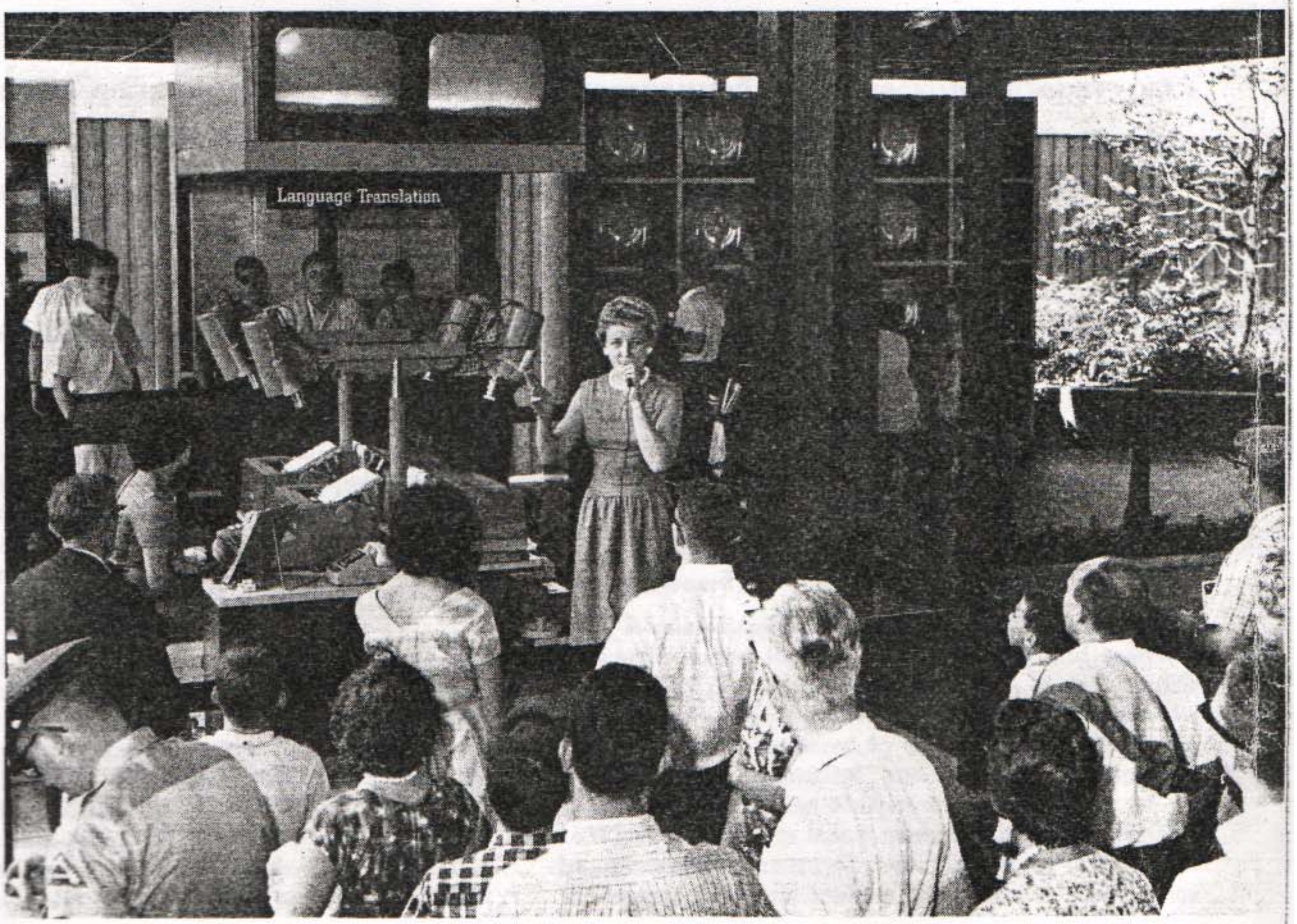


Fig. 1 Language Translation Exhibit at the IBM Pavilion, New York World's Fair.

The Russian-to-English machine translation system is one of the more interesting technological demonstrations at the New York World's Fair. Here, exclusively for Computer Design readers, is the first public information on the construction of the system, its dictionary organization, and its translating production operating mode.

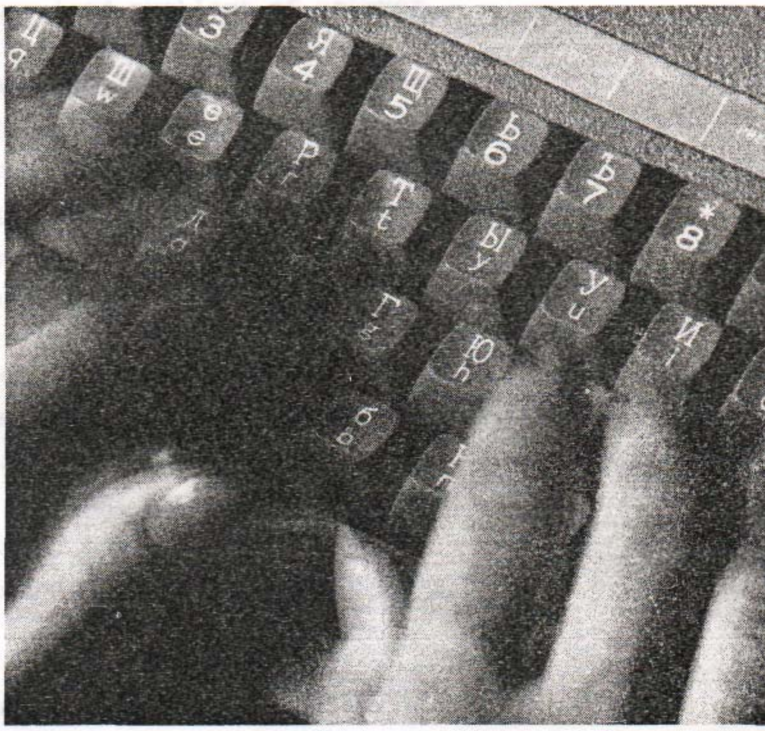


Fig. 2 Input typewriter showing Cyrillic-Roman keyboard.

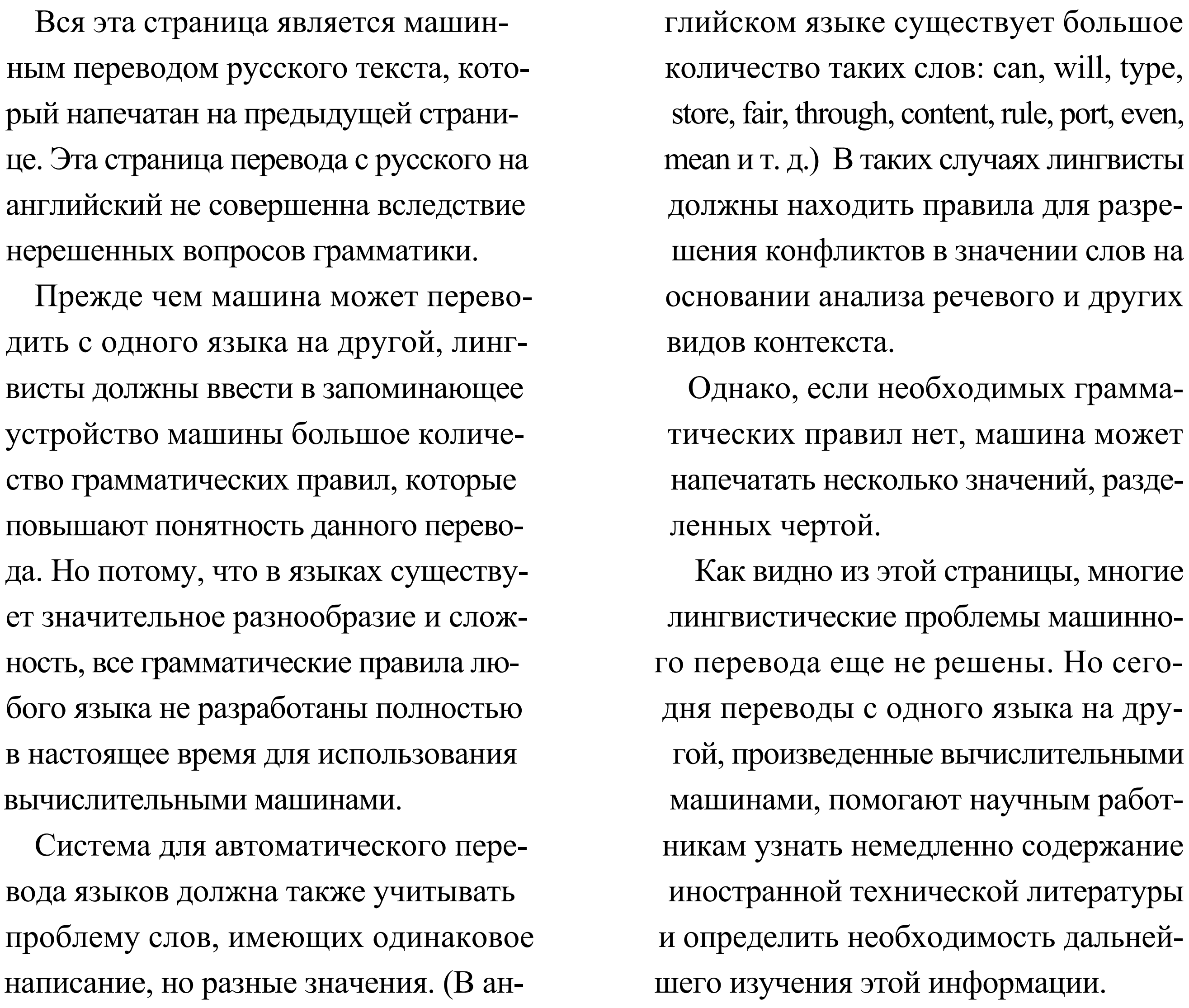
Visitors to the 1964-1965 New York World's Fair are exposed to a bewildering display of American ingenuity and technological progress. Much exhibit space is devoted more to showmanship than engineering enlightenment, but many exhibits present performances which demonstrate the state-of-the-art within their fields. One of these is an operating, demonstrating, producing electronic Language Processing System, capable of translating technical Russian into English: the IBM Research Language Processor.

The Language Translation exhibit in the IBM pavilion at the Fair grounds consists of two teleprocessing terminals and closed circuit television equipment to facilitate viewing by the audience as shown in Fig. 1. The input keyboard-printer of each terminal is fitted with typing keys and a Selectric "golfball" printing element which have the usual Roman alphabet and numerics in the lower case, but contain the Russian Cyrillic alphabet in the upper case (See Fig. 2). The output printer is equipped with the normal configuration — the Roman alphabet in both upper and lower case. An operator at the input keyboard types in a Russian word, sentence, paragraph, or a number of paragraphs, and the input is printed as it is typed. At the end of the desired unit of translation, a sign-off key (denoted "end-of-section") is struck and the input keyboard is automatically frozen, preventing the entry of further input. Almost immediately the output printer begins to type out the English translation of the Russian input; when the complete translation has been typed, the output printer stops and more input may be entered. An example of Russian input and the corresponding machine-translated English is shown in Fig. 3.

The operation of the Language Processing System in the demonstration mode may be followed by means of the block diagram in Fig. 4. The operator enters the Russian input on the keyboard at the Fair Site, and it is sent, character-by-character, over a telephone line to the Language Processing Laboratory at the IBM Kingston, New York location. The teleprocessor Control Unit at Kingston accumulates input characters until a full typewritten line is received, and the line is then transferred into the memory of the 1460 Data Processor. The 1460 accumulates lines until the full input section has been received, and upon receiving the end-of-section terminating character, transfers the entire section into the Lexical Processor. The Lexical Processor produces an English translation of the Russian text, and the translated section is returned to the 1460 Data Processor. The 1460 formulates lines for the Output Printer and transfers one line-at-a-time into the Teleprocessor Control Unit. This unit then sends one character at a time over the telephone lines to the Output Printer until the entire translated section has been printed. The Input Typewriter is then enabled and the next Russian input may be entered.

Machine Language Translation

The system which performs the translation for the World's Fair is a general-purpose language processor; that is, the hardware is designed to translate any source



Вся эта страница является машинным переводом русского текста, который напечатан на предыдущей странице. Эта страница перевода с русского на английский не совершенна вследствие нерешенных вопросов грамматики.

Прежде чем машина может переводить с одного языка на другой, лингвисты должны ввести в запоминающее устройство машины большое количество грамматических правил, которые повышают понятность данного перевода. Но потому, что в языках существует значительное разнообразие и сложность, все грамматические правила любого языка не разработаны полностью в настоящее время для использования вычислительными машинами.

Система для автоматического перевода языков должна также учитывать проблему слов, имеющих одинаковое написание, но разные значения. (В английском языке существует большое количество таких слов: can, will, type, store, fair, through, content, rule, port, even, mean и т. д.) В таких случаях лингвисты должны находить правила для разрешения конфликтов в значении слов на основании анализа речевого и других видов контекста.

Однако, если необходимых грамматических правил нет, машина может напечатать несколько значений, разделенных чертой.

Как видно из этой страницы, многие лингвистические проблемы машинного перевода еще не решены. Но сегодня переводы с одного языка на другой, произведенные вычислительными машинами, помогают научным работникам узнать немедленно содержание иностранной технической литературы и определить необходимость дальнейшего изучения этой информации.

Fig. 3 Sample of Russian-to-English machine translation.

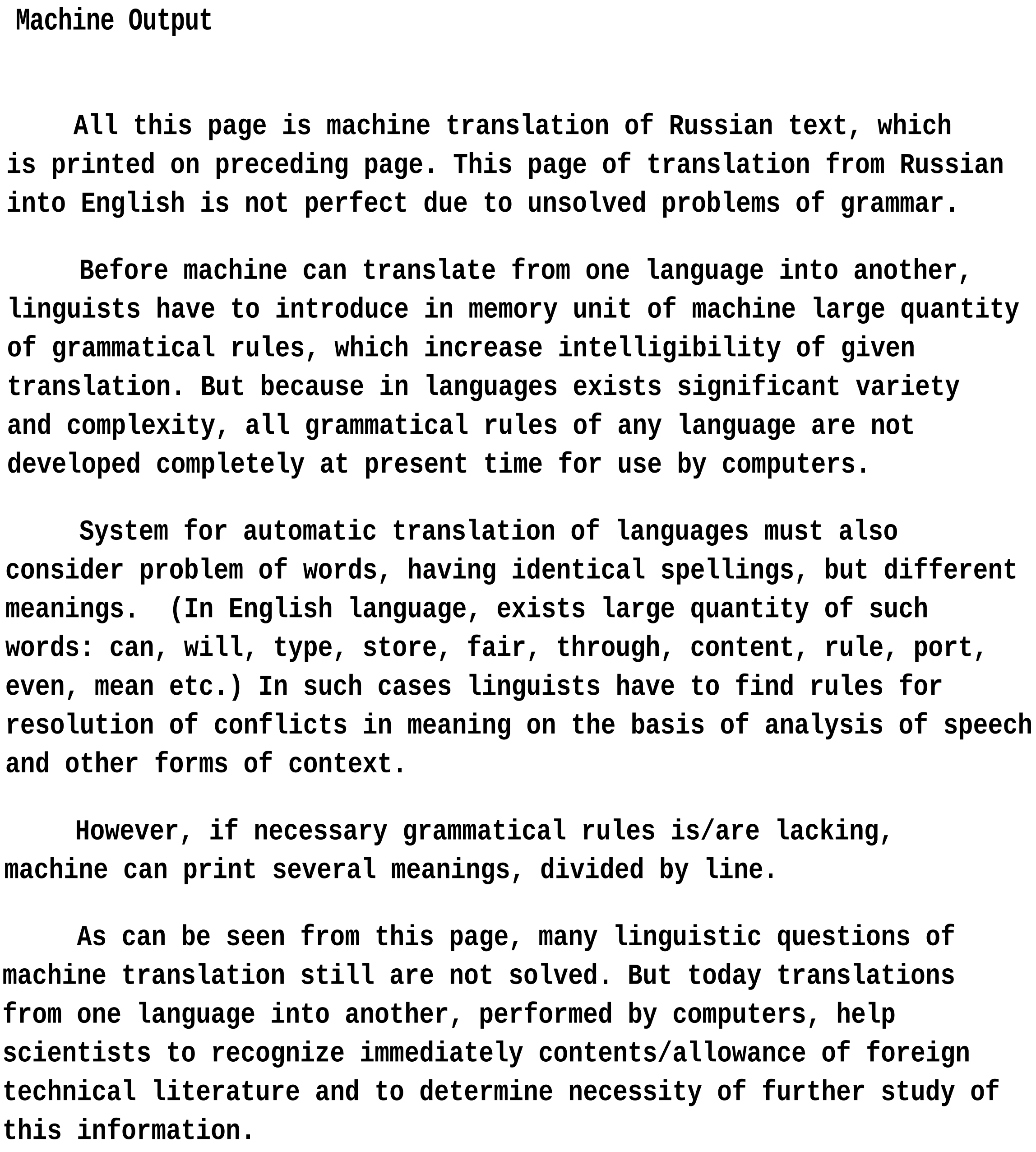
language into any target language, when provided with the appropriate dictionary and program. The hardware is also designed to be used in other applications of information retrieval and string processing, since it combines the features of a stored-program computer and a string processor. The discussion here will be limited to the World's Fair display, that is, the translation of Russian into English.

The system with its Russian-English program and dictionary produces a usable, but not perfect, translation, by providing lexical recognition of a large number of words, and syntactic analysis within a limited environment of each word. The mass memory required for storage of the 150,000 word Russian-to-English dictionary has a capacity of 65 million bits and an average random access time of 25 milliseconds. The translation program investigates up to 4 words to the left or right of each Russian word, for syntactic clues which will improve the translation. Words which cannot be found in the stored dictionary (e.g., proper names) are rendered phonetically into English.

The Dictionary

The Russian-English dictionary is basically an alphabetical listing of Russian words, ordered and searched high-to-low to insure that the longest (most significant) entry will be the first found; once it is found, the search may be considered to be successfully completed. The desired grammatical and semantic information, stored with the Russian word, is then extracted.

When a search is being made in the dictionary, all input characters are considered as one string upon which the longest possible dictionary match will be made, since



Machine Output

All this page is machine translation of Russian text, which is printed on preceding page. This page of translation from Russian into English is not perfect due to unsolved problems of grammar.

Before machine can translate from one language into another, linguists have to introduce in memory unit of machine large quantity of grammatical rules, which increase intelligibility of given translation. But because in languages exists significant variety and complexity, all grammatical rules of any language are not developed completely at present time for use by computers.

System for automatic translation of languages must also consider problem of words, having identical spellings, but different meanings. (In English language, exists large quantity of such words: can, will, type, store, fair, through, content, rule, port, even, mean etc.) In such cases linguists have to find rules for resolution of conflicts in meaning on the basis of analysis of speech and other forms of context.

However, if necessary grammatical rules is/are lacking, machine can print several meanings, divided by line.

As can be seen from this page, many linguistic questions of machine translation still are not solved. But today translations from one language into another, performed by computers, help scientists to recognize immediately contents/allowance of foreign technical literature and to determine necessity of further study of this information.

Fig. 3 Sample of Russian-to-English machine translation.

all input words and all dictionary entries are completely variable in length. Even word boundaries have no meaning for dictionary machine purposes, since the dictionary entry may be a phrase (e.g., “over the hill,” “state-of-the-art”). The high-to-low dictionary search is directed to start at a point higher than the entry which will ultimately be matched upon[1], and the dictionary entry found at that point is compared sequentially, character-by-character, against the string of input characters. If a mismatch occurs, that dictionary entry is discarded and the comparison of the next following dictionary entry with the same input string occurs. This procedure continues until a dictionary entry is found which matches completely with a portion of the input string; this condition is defined by the occurrence of a “match” indicator in the dictionary entry, just following the source language portion of the entry.

When a “match” occurs, the desired information is read out of the dictionary from the matching entry: the next search is directed to start, using the unmatched portion of the input string.

Note that the search procedure guarantees that the most significant dictionary entry (“longest match”) will be the one found. For example, “state-of-the-art” will be compared before “state” (since high-to-low ordering also implies long-to-short when one entry, “state-of-the-art,” happens to contain a shorter one, “state”) ; and. therefore, the entire phrase will be matched upon when it occurs in the input. When another phrase containing “state” (e.g., “state control . . .”) occurs in the input, the word “state” alone, will be matched upon.

Entries which could cause undesirable partial matches (e.g.. “stat”) are stored lower than “state,” and are, therefore, not discovered by the search algorithm.

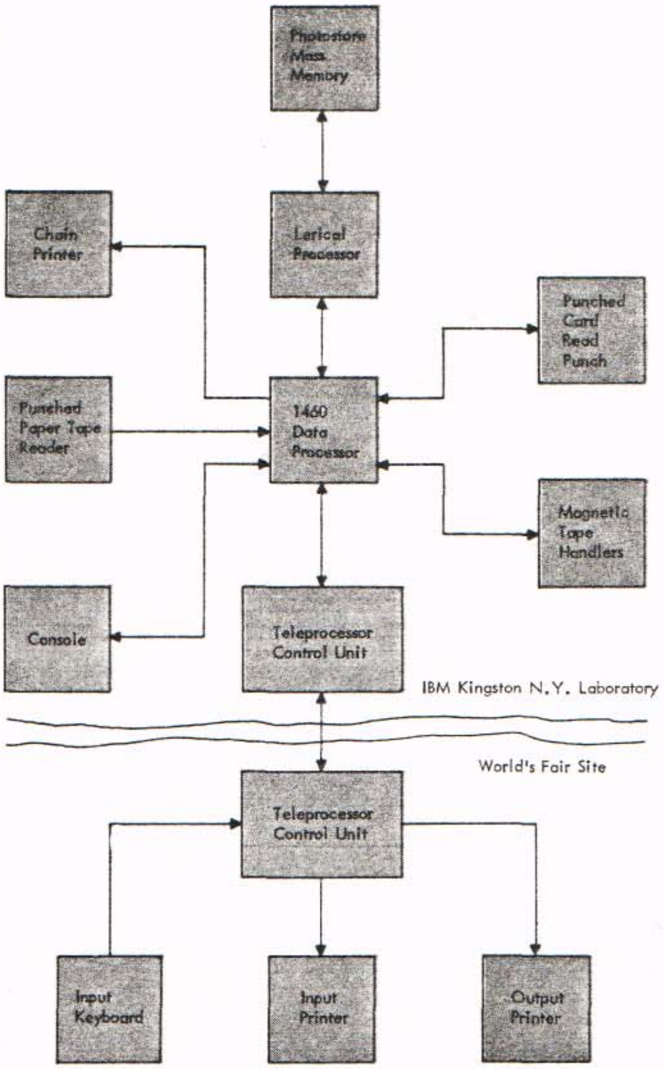


Fig. 4 Language Processing System.

There are pitfalls which must be considered and avoided. For example, a word not in the dictionary could be partially matched on a smaller constituent word, leaving meaningless syllables for the next match (e.g., "antic" matches on "ant" leaving "ic"). Careful dictionary structuring and the use of word boundaries (space, hyphen) are used in this Russian-English program to eliminate this problem. Also, many sophistications, e.g., tree structures, can be applied to the basic dictionary organizational concept; the foregoing serves only to describe the basic dictionary search.

The Dictionary Entry

Each dictionary entry is constituted as follows:

(prefix) (argument) (confix) (function)

In the pure case, the argument and function are, respectively, the source language and target language words of interest; i.e., given x (the argument, or source language word) find $f(x)$ (the function or target language word). The confix ("contextual prefix") usually contains coded grammatical information pertinent to the argument; this confix will become a prefix for the next search; the next search, then, would match upon a prefix-argument string of symbols. In this way, grammatical information is carried through the word-by-word dictionary search. Not only is the syntactic information in the confix-prefix operation carried from left-to-right through the sentence, but the left-to-right motion may be stopped or even reversed by instructions contained in the dictionary entry. There are, in fact, several types of "match" indicators; one continues the left-to-right motion; another instructs a rematch on the same input stream, using the new prefix information obtained from the match on this entry. Prefix-confixes may be modified, erased, or added to during a match.

To illustrate the necessity for this alteration of the normal left-to-right motion of the translation program (which alteration confers the name of "bidirectional" upon the program), consider the two-word Russian phrase: С ГРУППАМИ which means "with groups". Eighteen dictionary searches are required to translate these two words, because the translation of С depends upon the case of the noun which is its object. Therefore, the translation program in its normal left-to-right motion through the input data, must look up С, discover (through its confix) its peculiarities, look up the stem ГРУПП and the ending АМИ, to discover that ГРУППАМИ is the instrumental plural: then it must go back and look up С again and choose the translation "with" (had the noun been in the genitive, С would have been translated "from"): finally it must translate ГРУППАМИ as "groups".

Language Processor: Dictionary Organization

The process of machine translation consists wholly of a series of searches of a number of lists. The normal machine instructions and the data upon which the machine is operating are freely intermixed in dictionary entries. This, of course, is contrary to conventional stored program machines where instructions and data, although free to operate upon one another, are kept separate, and the instructions are stored and executed primarily in a predetermined order.

In the Language Processor (LP), the order of execution of instructions is determined by the input data. An initialization procedure exists for getting started once new input has been loaded (look at the first input character, find — from a directory — where to start the comparison with the dictionary). From there on, the sequence of LP operations is determined solely by what dictionary entries are matched upon, and what instructions they contain. Frequent restarts through references to the directory are made, programmed by dictionary entries. Using this organization, the entries consist of three basic types:

1. Normal-dictionary-type entries which give grammatical information about a word, stem, ending, phrase, plus its translation.

2. Data processing-type entries which contain instructions related to grammatical processing, e.g., forward or backward skips.

3. Data processing-type entries which have nothing to do with grammar, but contain instructions related to data processing, e.g., initialization, end-of-translation routine, load input, dump output.

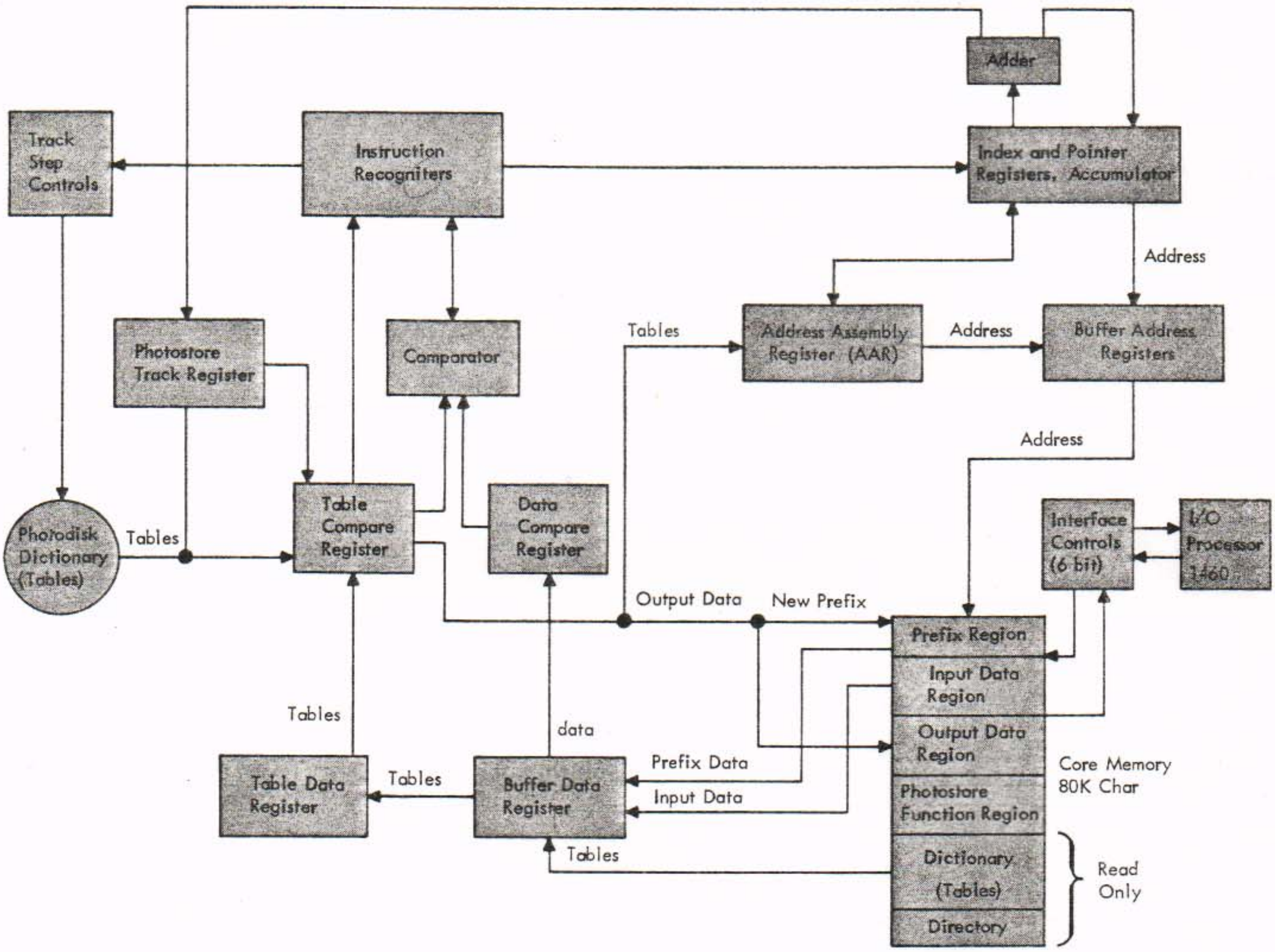


Fig. 5 Functional block diagram of the Language Processor (LP).

The entries of types 2 and 3 are called "control" entries, and those of type 1 are called "dictionary" entries. The union of all entries is termed the LP "tables." In the World's Fair tables, there are approximately 125,000 dictionary entries, and 25,000 control entries.

Of these 150,000 entries, there is a wide disparity in the frequency of use of each. Some are matched on quite frequently (e.g., initialization, common Russian words) while others are infrequently used. For the purposes of storage in the LP, therefore, the tables are divided into two parts. Approximately 3500 entries which are the most frequently used are stored in a high-speed (2.5 usec) core memory, where average random access time to a particular entry is less than ½ ms. (The reason for a 500 usec average random access time to an entry in a 2.5 usec core memory becomes apparent if one reviews the previously-described searching process through which the desired entry is located.) The remaining entries are stored on a photo-disc mass memory. On the average, 9 dictionary matches are required for each Russian word processed. Using the technique of splitting the tables between high-speed, low capacity core, and (relatively) slow-speed, high capacity disc, 75% of all matches will occur in the core tables. A translation speed of about 20 words-per-second can be achieved with this method. If all entries were stored on the disc, the speed would be about 4½ words-per-second.

This dual table organization introduces the requirement that the LP begin each dictionary search in the core tables, find a match, and determine (from instructions contained in the matching core entry) whether or not there is a possibility that a more significant match might be found in the disc. If this possibility exists, the search must continue on the disc. Therefore, all searches start in core, continuing on the disc if necessary.

LP: Functional Operation

The LP is functionally-diagrammed in Fig. 5. The core tables (including a directory) are stored in core, and the photodisc tables exist in the Photostore mass memory. Areas of core are reserved for storages of prefixes, input data (Russian), output data (English), and photostore functions (described later). A section of input Russian is placed into the input data region, and pointer registers are set to indicate the core location of the part of input Russian to be operated on, the core location where any output English is to be written, the core location where the first table entry is to be read for comparison

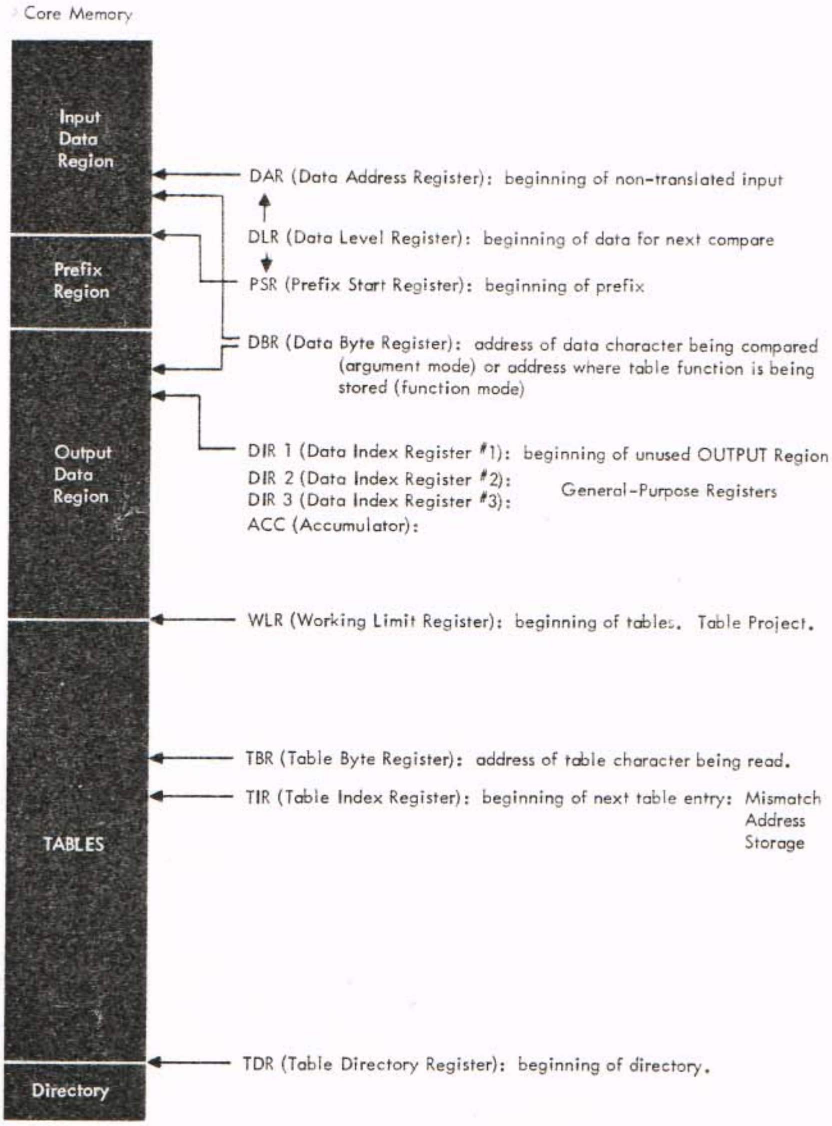

Core Memory
Input Data Region
Prefix Region
Output Data Region
TABLES
Directory
DAR (Data Address Register): beginning of non-translated input
DLR (Data Level Register): beginning of data for next compare
PSR (Prefix Start Register): beginning of prefix
DBR (Data Byte Register): address of data character being compared (argument mode) or address where table function is being stored (function mode)
DIR 1 (Data Index Register #1): beginning of unused OUTPUT Region
DIR 2 (Data Index Register #2):
DIR 3 (Data Index Register #3):
ACC (Accumulator):
General-Purpose Registers
WLR (Working Limit Register): beginning of tables. Table Project.
TBR (Table Byte Register): address of table character being read.
TIR (Table Index Register): beginning of next table entry: Mismatch Address Storage
TDR (Table Directory Register): beginning of directory.

Fig. 6 LP Pointers and Registers.

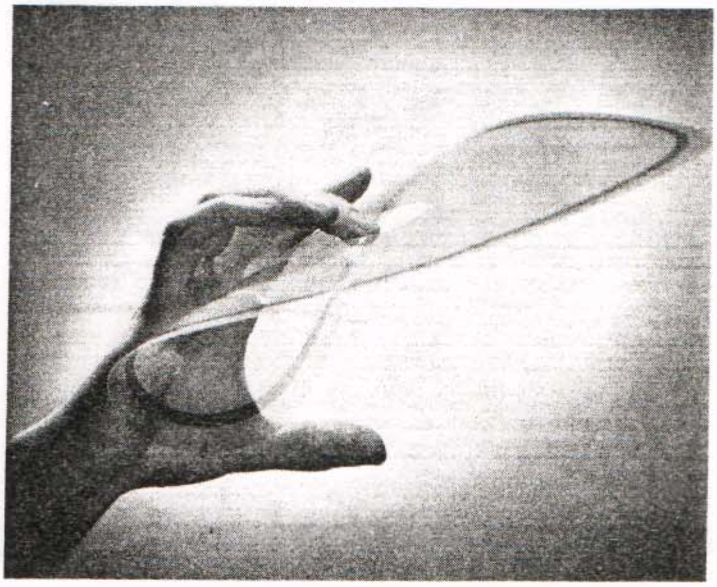
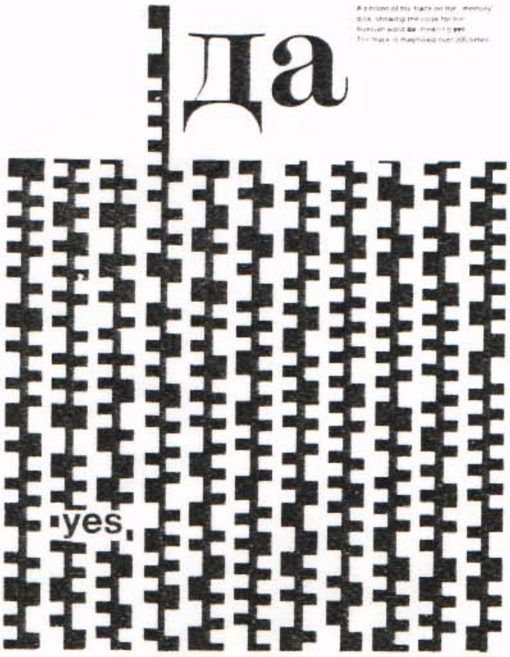


Fig. 7 Photoscopic rotating disc is the basic storage element of the Photostore memory.



A portion of the track on the "memory" disk, showing the code for the Russian word да meaning yes. The track is magnified over 200 times.

Fig. 8 Photostore data tracks.

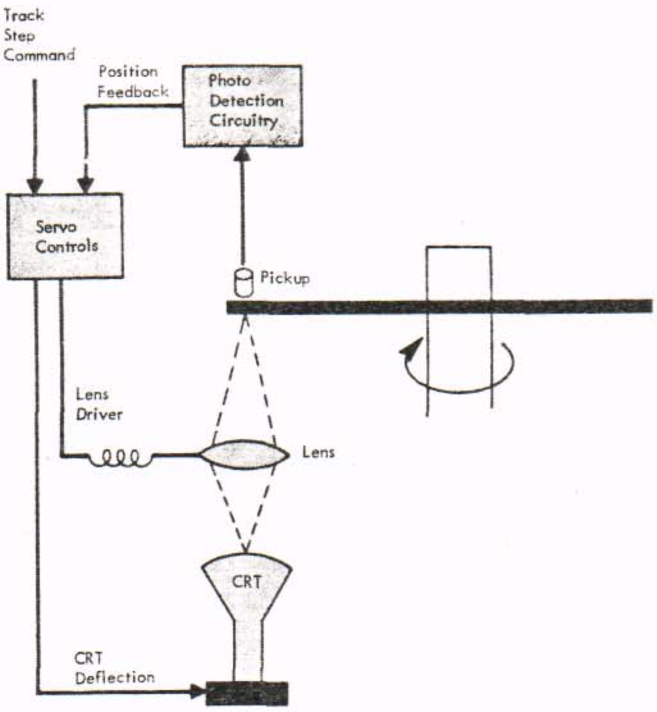


Fig. 9 Photostore beam positioning mechanism.

(i.e., where the table search starts), and the core location where the prefix to the input Russian is to be found. The following sequence of events is then faithfully and repetitively executed by the LP:

1. Form a data stream from the prefix data followed by the input data, and feed it through the Buffer Data Register into the Data Compare Register.

2. Read the table stream starting at the pointer-designated core location, and feed it to the Table Compare Register.

3. Compare table argument with data stream, character-by-character; so long as they match, execute any instructions in the table stream as they occur.

4. If a mismatch occurs before a "match" instruction is encountered in the table stream, return to step 1 and repeat the comparison using the same data stream, but the next sequential table entry. Note that the pointers which designate the core locations of the data and table characters being fed to the comparator have been moving through their respective streams; therefore, the data pointer must be reset to start from the original location, and the table pointer must skip to the beginning of the next entry. The address of the next sequential entry is contained in the beginning of each table entry (the "mismatch address," or "chaining address"). It is stripped off by the Address Assembly Register and stored in an index register while the data vs. table comparison proceeds. When a mismatch occurs, this "chaining" address is transferred to the table pointer.

5. If no mismatch has occurred by the time a "match" instruction is encountered in the table stream, a successful match has been made on the argument of that table entry.

The LP now changes from "Argument" (compare) mode, to "Function" (readout) mode. The confix of the table entry is stored into the prefix region, to become the prefix for the next comparison. The function of the table entry is stored into the Output Data Region. All pointers are moved up to prepare for the search on the next unit of input, and the writing of the next function into the next locations of the output region. The automatic movement of these pointers can be (as seen in С ГРУППАМИ example) changed by instructions in the table entries. After each successful match, one of two events is programmed: the following search will either start fresh through the directory, or will start at some specific table entry, whose address is contained in the matching entry.

The above (somewhat simplified) sequence of events is clear so long as the matching table entry exists in the core. Whenever it is necessary to search for a disc table entry, the normal core search will match on a core table entry whch contains instructions whose sense is as follows:

A match on this table entry argument has been made. There may be a longer match on the disc. Search on the disc, starting at track #XYZ. If a more significant match is found there, use that entry. If not, use this entry as the match.

Referring again to Fig. 5, when a disc search is found necessary, the Photostore Track Register, which always contains the number (address) of the track currently under the reading mechanism, is compared with the desired track *XYZ*, stored in the Data Compare Register. A track step signal is generated to move the reading mechanism toward track *XYZ*; this process is continued until the desired track is reached. The disc entries are then searched in the usual high-to-low fashion, and if the more significant match is on the disc, the function portion of the entry must be read from the disc (at a 2.7 mc bit rate) immediately following the match indicator of that entry. This function is temporarily stored in the Photostore Function Region of core until it can be properly executed.

Fig. 6 shows the pointer registers used in the LP to keep track of the data and table streams, and indicates their functions. Instructions exist to transfer any of these registers to any other, to increment or decrement any register, and in general, to manipulate the contents of these registers and the memory locations which they address. Through such manipulations, the entire processing capability of the Language Processing Equipment is achieved.

The Photostore

The heart of the Photostore[2,3] is a rotating disc on which the coded table information is recorded (see Fig. 7). All information is recorded in an annulus of approximately 0.4 inch at the outer edge of the disc. Within this annulus are 1000 concentric tracks of photographic information; each track stores about 8200 eight-bit characters, making a capacity of about 65 megabits.

A magnified cross section of several tracks is shown in Fig. 8. A track of information is bordered by a dark band and a light band, and a binary cell consists of a light spot and a dark spot; the order in which the dark and light spot occur determines whether a cell contains a binary ONE or a ZERO. The information is read from the disc using a light beam, the diameter of which is of the dimensional order of the information track,

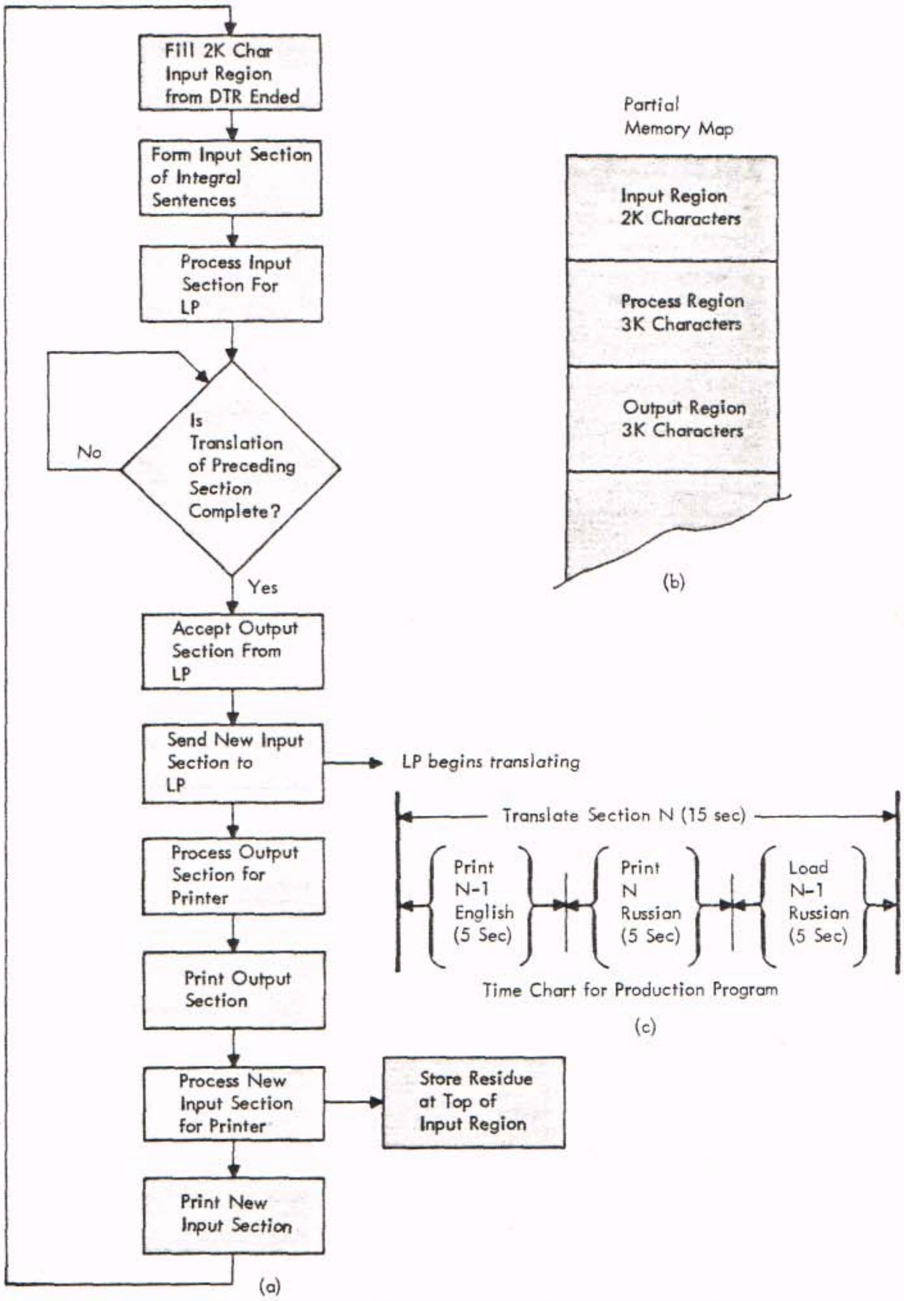

Fill 2K Char Input Region from DTR Ended
Form Input Section of Integral Sentences
Process Input Section For LP
Is Translation of Preceding Section Complete?
No
Yes
Accept Output Section From LP
Send New Input Section to LP
LP begins translating
Process Output Section for Printer
Print Output Section
Process New Input Section for Printer
Store Residue at Top of Input Region
Print New Input Section
(a)
Partial Memory Map
Input Region 2K Characters
Process Region 3K Characters
Output Region 3K Characters
(b)
Translate Section N (15 sec)
Print N-1 English (5 Sec)
Print N Russian (5 Sec)
Load N-1 Russian (5 Sec)
Time Chart for Production Program
(c)

Fig. 10 Overlap in the Production Translation mode.

and photodetection circuitry. The light beam is generated by a cathode ray tube, and positioned both by deflection of the CRT beam, and by movement of the lens which concentrates the beam. Feedback information is generated by the photodetection circuitry, since movement of the beam off the track will move it either into the dark or light boundary areas, creating an average light condition which is markedly different from the equal areas of dark and light encountered on the track. The beam positioning hardware is diagrammed in Fig. 9.

The disc rotates at a speed of 2400 rpm, creating an output data rate of 2.7 megabits per second. This serial data stream is divided into eight-bit characters, resulting in an output character rate of 325,000 per second, which can be accommodated by the 2.5 usec memory in the LP.

Data Processing Equipment

Expression of two points of view may help to explain the relative roles of the Language Processing Equipment and the Data Processing Equipment in the World's Fair Translation System. To the Language Processor, the Data Processor exists only to relieve the LP of the bothersome and routine tasks associated with input/output control and processing. To the Data Processor, the Language Processor is only another unit of peripheral equipment which must be controlled and accommodated.

The interface between the LP and the 1460 is not different from many other processor-peripheral interfaces. Each unit can raise an interrupt line to request communication with the other. When the unit called has acknowledged the request, the calling unit indicates the nature of the request, on data transfer lines. The two units then proceed with the requested operation, which is always to transfer information, a character-at-a-time, from LP to 1460 or from 1460 to LP.

The Demonstration Mode

The following requirements are placed upon the 1460 in the handling of the World's Fair demonstration:

1. Accept input from and control the teleprocessing equipment, as previously described.

3. Prepare each section for processing by the LP. The two tasks involved here are code transformation and code expansion. Code transformation is required because the separate requirements of the teleprocessing equipment and the LP dictionary result in the use of two entirely different code sets. Code expansion of certain characters is done in order to provide a common character preceding certain subsets of characters. (The benefit of this particular code expansion is that all characters which may follow a Russian word acquire the same initial character; therefore, the last character of every Russian word in the data input is followed by this initial character. This creates a delimiter which marks the end of each input word. Note that this circumvents one of the previously-mentioned "pitfalls.")

3. Send the transformed input section to the input region of the LP; when the LP signals that translation is complete, accept the translated section from the LP output region.

4. Prepare the output section for printing by the teleprocessing equipment. Again, code transformation is required.

5. Form lines for the Output Printer, and send the section one-line-at-a-time to the Teleprocessor Control Unit.

Production Translation Mode

The high-speed translation capabilities of the LP are not utilized in the demonstration mode, since the Teleprocessing Terminal Equipment limits the data transmission rates to 16 characters per second. Consider a typical two-sentence section consisting of 240 characters. This section can be input-processed by the 1460, translated by the LP, and output-processed by the 1460 in less than 2 seconds, but 15 seconds are required to send the resulting translation back to the output printer. To fully utilize the capabilities of the Language Processing Equipment, a Production Translation mode is used.

For this mode, input Russian is prepared on punched paper tape. The paper tape is read by a high-speed (500 characters-per-second) paper tape reader, under control of the 1460 (a magnetic tape input option also exists). The input text, which can be of any length, is sectioned by the 1460, processed, sent to the LP for translation, and printed in Cyrillic on the high-speed (272 lines-per-minute) Chain Printer. The output text (translation) is received from the LP, processed, and printed in English on the Chain Printer. More input is then accepted. The printed output thus contains the original input, in Cyrillic, followed by its English translation. (A magnetic tape output option also exists.)

In order to maximize the translation speed of the overall system, the various operations performed in production translation are completely overlapped. A gross flow diagram of the overlapped operation is presented in Fig. l0a. A memory map showing the areas of interest appears in Fig. l0b.

The 2000-character input region is loaded from the paper tape reader. The translation program considers the end-of-a-sentence to be a boundary, across which grammatical information is not carried and skips are not implemented. The 1460 is, therefore, required to send input text to the LP in sections made up of integral sentences. To accomplish this, the 1460 scans the newly-loaded input region from back-to-front, searching for a sentence termination (e.g., period, question mark, etc., followed by two spaces), and uses such a termination to divide the input region into two parts. The first part (the input text section) is processed for the LP (code conversion, character expansion) and placed into the "process region" of core (to allow for character expansion, this region is made larger than the input region). The second part (the residue) will later be placed at the top of the input region, since it is the beginning of the next input section. Moving of the residue must be delayed until the input section at hand has been completely disposed of. The situation now is as follows:

1. The raw nth input section is in the input region, along with the residue for the *(n + l)* input section.

2. The processed-for-LP nth input section is in the process region.

3. The *(n — l)* input section is in the LP. being translated.

At this time, translation of the *(n — l)* input section should be completed; the 1460 accepts output section, *n — l,* from the LP, moves input section, *n,* into the LP from the process region, and the LP begins translating input section, *n.*

The Cyrillic printout of section, *n—l,* has been performed in previous program cycle: the corresponding output must now be printed. Therefore, output section, *n—l,,* is processed for the printer and placed into the process region, replacing the processed-for-LP section, *n,* which is no longer needed since it has been sent to the LP. Output section (English), *n — l,* is then printed.

The Cyrillic printout of input section, *n,* is accomplished next. The raw input section, *n,* is processed for the printer and stored in the process region (replacing the no-longer-needed processed-for-printer output section, *n — l),* and then printed. The program is now ready to accept new input for the (*n +1)* section. A time chart showing one typical overlapped cycle is shown in Fig. 10c.

Supporting Operations

A number of off-line supporting operations exist to prepare the tables which are the heart of the translation operation. A master magnetic tape file consists of both pure dictionary and control entries. The processes of modifying the translation program in order to improve translation quality, and of adding words to the dictionary to increase the scope of input text which can be handled, are perpetually-continuing activities. Each time a block of dictionary improvements is accumulated, resulting in a group of new entries, these must be merged with the existing dictionary, maintaining the most-significant-match-order, new chaining addresses must be generated for those entries which will be contained in core, and the remaining entries organized for writing onto a new disc. The result of these operations are: (1) a magnetic tape, the contents of which will be loaded into the LP core table area via the 1460, and (2) a set of magnetic tapes, the contents of which will be used as input to the digital photographic equipment which manufactures a new disc. These supporting programs are. in general, performed on a 7044 or 7094.

Other LP Applications

The Language Translation complex created for demonstration of Russian-to-English translation at the World's Fair is designed to perform and is capable of performing a variety of other tasks. Machine translation of Chinese-to-English has been successfully demonstrated[4]. An operational program is in use, on a production basis, for the translation of Stenotype language to English[5]. Considerable progress has been made in French and German[6]. Processing of English is also being studied to use this kind of system for information processing applications where a large volume of input text must be machined-processed (cataloguing and indexing of books and periodicals to libraries, search of legal statutes, search of patent applications, and comparison with the patent file)[6]. Another obvious kindred application is the information retrieval problem which has much in common with the language translation problem.

END

REFERENCES

1. King, G. W., "Table Look-up Procedures in Language Processing: The Raw Text"; IBM Journal of Research and Development, Vol. *5.* No. 2, April, 1961.

2. Craft, J. L., E. H. Goldman, W. B. Strohm, "A Table Look-up Machine for Processing of Natural Language"; IBM Journal of Research and Development, Vol. *5,* No. 3, July, 1961.

3. King, G. W., G. W. Brown, L. N. Ridenour, "Photographic Techniques for Information Storage," Proceedings of the IRE, 41, No. 10, October, 1953.

4. King, G. W. and H. W. Chang, "Machine Translation of Chinese," Scientific American, June, 1963.

5. Galli, E. J., "The Stenowriter — A System for the Lexical Processing of Stenotype," IRE Transactions on Electronic Computers, Vol. EC-11, No. 2, April, 1962.

6. Goldman, E. H., "Automatic Translation," Proceedings of MESUCORA 63, Congres International, Paris, November 21, 1963, tome 2, seance no. 11.