Some of the Engineering Aspects of the Machine Translation of Language

ROBERT E. WALL, JR.
ASSOCIATE MEMBER AIEE

THE MECHANICAL translation of language by digital data-processing equipment was first suggested in private correspondence between A. D. Booth and W. Weaver in 1946.[1] Since that time, others have been attracted to the field of mechanical translation and their work has resulted in the publication of a magazine, a book, and numerous papers.[2,3,4]

At the time that mechanical translation was conceived by Weaver and Booth, the necessary computer components were not available to realize a mechanical translator. In this paper it will be shown that the recent advances in computer technology have made mechanical translation a definite possibility provided certain limitations are made in the type of material which will be fed into the machine, and also if certain allowances are made in the required quality of the output material.

The intent of this paper is to acquaint engineers with the general "state of the art" of the machine translation problem and to indicate some of the more promising methods of attack on the individual problems. The individual problems are not considered in detail. A careful analysis of either the coding problem or the memory requirements would require more space than is used for this entire paper. Detailed discussions will be handled in subsequent papers.

General Considerations

The people of every highly civilized speech community have a rich literary heritage. It would not seem reasonable to expect the first translating machines to translate this literature from one language to another and to preserve completely the original style and the more delicate shades of thought and meaning. There is, however, a very heavy demand for translations of scientific material at present. Since scientists tend to use a simple language structure in their writing, and since readers of scientific material are interested mainly in the intelligibility of the material, it is possible to make many useful restrictions on the amount of processing which a machine translator would have to perform if the input is restricted to scientific writing. For example, it has been shown[5] that Russian scientific writers tend to follow English word order form. It is likely that a machine translator can produce satisfactory Russian-English translations without rearranging word order in the "target" language. The "target" language is defined as the language into which the translation is made, while "source" language refers to the language from which the translation is made. The fact that only intelligibility in the output material is required allows a considerable number of restrictions in the number of target language words stored in the dictionary. For example, a Russian-English dictionary[6] lists with the word "окаоянство" the English equivalents "impiety, impiousness, ungodliness, godlessness." Nothing essential to understanding would be lost if only one of the four equivalents were stored. It might also be advisable to simplify the spelling of English words to decrease storage space, for example by storing "thru" instead of "through."

It seems unlikely that the quality of the translations from a machine would be equal to that supplied by a well-trained human translator. The machine would have to justify its existence by superiority over the human in speed and, if possible, by a lower per-word cost of translation. A language translation machine would undoubtedly represent a very large investment since it would require a large temporary storage space to allow the necessary processing of material, and it would have to have a permanent memory with a very large capacity and a relatively low access time. It seems likely that such a machine will cost as much as a modern large-sized digital computer which has a per-hour cost of about $250. Human translators will produce rough, understandable translations for a per-word cost of about one cent. This means that a $250-per-hour machine would have to produce about 25,000 words per hour or, in round numbers, about ten words per second, in order to have the same per-word cost. This figure is, of course, somewhat arbitrary, but it is certain that 1 word per second is too slow, and 100 words per second is probably faster than necessary.

Translation Process

Four definite steps are required in the translation process: encoding, memory search, logical operations, and decoding. Consider these four steps in order.

ENCODING

Two basic problems arise when an attempt is made to encode text material into the machine language, or "code." The first problem is that of the optimum selection of the machine code, and the second is that of transcribing printed text material into machine code. Printed material could be converted into machine code in many ways. Operators might transcribe the text material onto punched or magnetic tape; an electronic tape reader could then read the tape and supply the necessary electrical pulse patterns to the machine. As another possibility, the duties of the human transcriber and the mechanical tape reader might both be performed by an all-electronic device. A machine has been recently announced which will read and encode English at the rate of 3,600 words per minute.[7] It should be possible to modify this machine so that it will read and encode the Russian Cyrillic alphabet. It is beyond the scope of this paper to discuss the configuration of an electronic encoder in detail; however, such a device is very important for the realization of economical machine translation since human transcribers who know only English would copy foreign language material slowly and there would be a high probability of error.

The machine code should be selected in such a way that the code groups were of uniform length, and so that storage of material would require the least number of bits. It appears that it is necessary to provide both capital and lower-case letter codes in order to process the material properly. The total requirement for Russian will then be about 80 distinct code groups: 9 for the number codes, about 10 for punctuation marks, and 61 for letter codes (5 of the 33 Russian letters never start a word and will not

Paper 56-693, recommended by the AIEE Computing Devices Committee and approved by the AIEE Committee on Technical Operations for presentation at the AIEE Summer and Pacific General Meeting, San Francisco, Calif., June 25-29, 1956. Manuscript submitted March *27*, 195G; made available for printing May 4, 1956.

ROBERT E. WALL, JR., is with the University of Washington, Seattle, Wash.

Gratitude is expressed to Profs. Laurel J. Lewis, W. Ryland Hill, and Erwin Reifler for many helpful criticisms and discussions.

Table I. Proposed Code

Symbol	Lower Case Code	Capital Code
а	00001	1111100001
б	00010	1111100010
в	00011	1111100011
г	00100	1111100100
д	00101	1111100101
е	00110	1111100110
е	1111011110	
ж	00111	1111100111
з	01000	1111101000
и	01001	1111101001
й	1111111010	
к	01101	1111101010
л	01011	1111101011
м	01100	1111101100
н	01101	1111101101
о	01110	1111101110
п	01111	1111101111
р	10000	1111110000
с	10001	1111110001
т	10010	1111110010
у	10011	1111110011
ф	10100	1111110100
х	10101	1111110101
ц	10110	1111110110
ч	10111	1111110111
ш	11000	1111111000
щ	11001	1111111001
ъ	1111000010	
ы	11010	
ь	1111000011	
э	1011	1111111011
ю	11100	1111111100
я	11101	1111111101
1	1111000100	
2	1111000101	
3	1111000110	
4	1111000111	
5	1111001000	
6	1111001001	
7	1111001010	
8	1111001011	
9	1111001100	
Space	00000	
Comma	1111001110	
Semicolon	1111010000	
Period	1111010001	
etc.		

need to be capitalized). A demand for 80 distinct code groups would seem to require a 7-bit code, but it will be shown that a basic 5-bit code may be made to yield 80 distinct code groups and, if properly selected, the 5-bit code will result in a considerable saving in storage. An example of such a basic 5-bit code is displayed in Table I. To test this code a random sample of a half-page of Russian text was analyzed; the occurrences of the various symbols are listed in Table II. The sample had a total of 786 symbols. Coding this material with a 7-bit code would require 786X7 totals 5,502 bits. If the code shown in Table I were used, there would be 745 occurrences of 5-bit codes and 41 occurrences of 10-bit codes; see Table II. The total number of bits required would be 41X10+745X5 = 4,135 bits, or an average of 4135/786 = 5.23 bits per symbol. Thus a considerable saving in storage space results from using this modified 5-bit code over that required with a 7-bit code.

Many other codes could be devised. The illustrated code does have a uniform code-group length. It was shown that a random sample of text could be coded with an average code length per symbol of only 5.23 bits; this is only 75 per cent of the bits required with a 7-bit code.

MEMORY SEARCH

The permanent memory must also store essentially all source language word sequences (idioms) which do not give understandable word-for-word translations into the target language. Each source language entry must have the necessary target language equivalents stored with that entry in the memory. During the memory search, incoming text words may be compared with the source language entries in the memory, first in groups of words, with the idiomatic memory, then by single words, with the individual-word memory.

It has been shown[8] that words of rare occurrence must be translated, since very often these words determine the meaning of the whole sentence. Therefore, it would seem necessary to store the complete source language-target language dictionary, with the exception of internationals. The largest German-English dictionary[9] contains approximately 400,000 entries. When the selection of English equivalents is made it should not be necessary to store more than an average of two equivalents for each source language words. If internationals are not stored, the total storage requirement is probably about 1,000,000 words. A reasonable estimate of the average length of a word would be 7 letters, or a little more than 35 bits using the code of Table I. The individual-word memory would thus contain about 40 megabits of information. The storage requirement for the idiomatic memory is considerably less, definitely not over 60,000 words in total.

The Russian language does not have as many words as either German or English but the storage requirements are probably about the same because in Russian 59 word affixes[10] are used to denote the grammatical usage of a word. The Russian noun has six cases and may be either singular or plural, and an adjective must agree with the noun that it modifies both in gender (there are three genders in Russian) and in case.

Table II. Occurrence of Russian Letters, 1/2 Page of Text

Symbol	No. of Times Occurring
а	35
б	5
в	41
г	7
д	12
е	57
ё	4
ж	2
з	12
и	79
й	6
к	24
л	26
м	18
н	50
о	63
п	19
р	42
с	36
т	37
у	15
ф	10
х	14
ц	11
ч	12
ш	1
щ	2
ъ	0
ы	17
ь	4
э	5
ю	5
я	17
Space	80
Comma	11
Colon	0
Semi colon	0
Period	4
Nos. and Symbols	5
Capitals	5

The amount of material stored in the memory system should be kept as small as possible, and the material should be stored in the memory in such a way that the search may proceed in the most efficient manner. It is possible to decrease the amount of material stored in the memory by not storing internationals. Other schemes for decreasing dictionary size depend on reducing a word to simpler components, such as reducing a compounded word to the words of which it consists, or dissecting a word into a stem and an ending in the case of affixed words. Eliminating the storage of internationals presents no serious logical difficulties per se. If an incoming text word should not correlate with any of the source language words in the dictionary, the word could be printed out in its original form or in its target-language phonetic equivalent. It is necessary to make a provision for printing out uncorrelated words anyway, in order to take care of misspelled words. Routines which involve dissection schemes would give a considerable decrease in memory size, but they all pre-

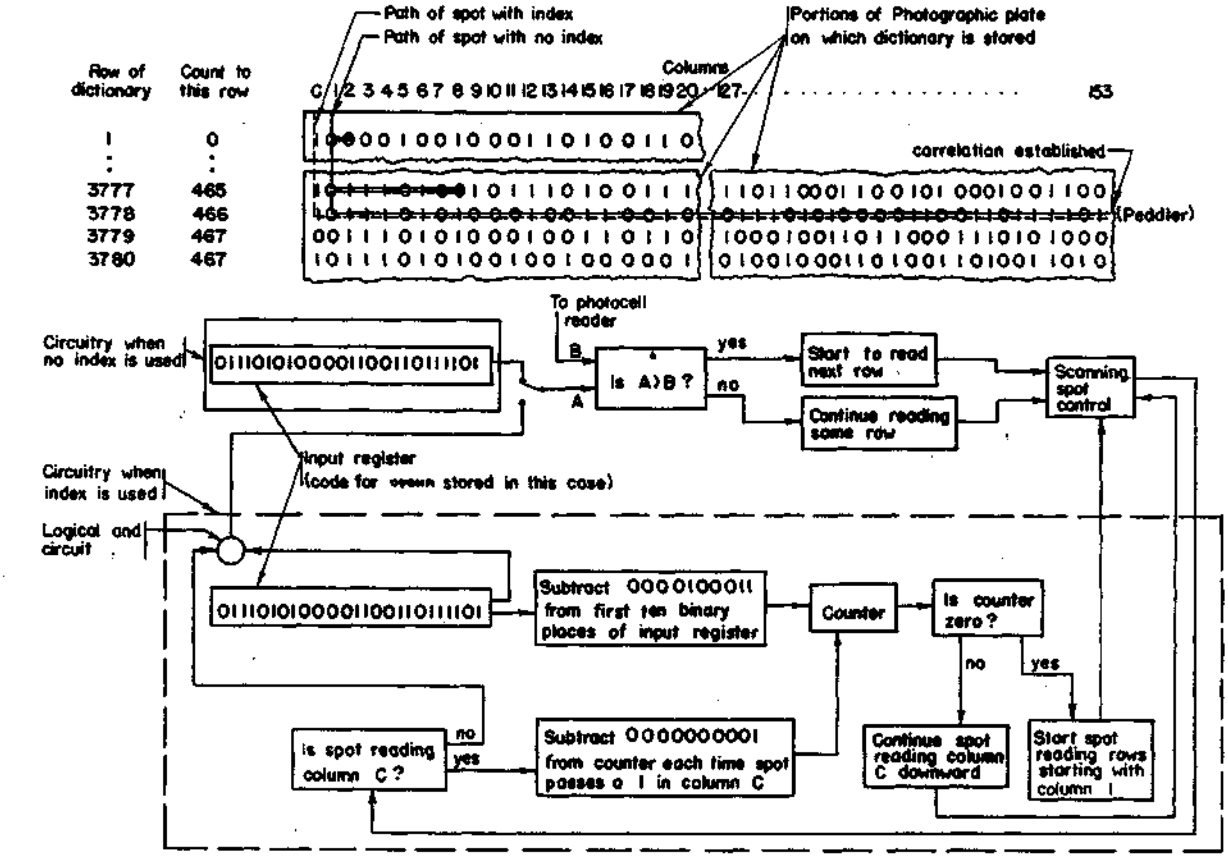

Fig. 1. Block diagram of optical memory

sent logical difficulties in their complete form. For instance, if separate stem and ending dictionaries are provided, difficulties result from the many irregularities of language and the necessity of coordinating the two dictionaries. It would seem useful to break up a word into stem and ending within the individual dictionary entry by listing the word stem first then the endings along with the target language equivalents. An incoming text word may then be compared with a stored source language word starting with letters farthest to the left in the words. If correlation is established, the target language equivalents stored with the correlated ending are read out and stored to await further processing.

A scheme for dissecting compounds has been developed by Reifler.[11] If a word is not located in the dictionary, the letter at the end of the word is ignored and the remainder of the word is compared with the dictionary. If this remainder is not located, another letter at the end is ignored. This process is continued until the remainder is located or until all the letters have been used up. Then the machine starts to ignore letters from the front of the word and to compare the remainder with the dictionary as before. When a portion of an incoming word has correlated with a stored sequence, then the ignored portion of the word must also be compared with the dictionary. If letters are ignored from the front of the word, or if ignored portions of a word are to be compared, it often would be necessary to go to a part of the memory which is far removed from the place that was searched in the previous step. This skipping about in a large memory system is a very serious problem. Any word which could not be correlated with a word in the dictionary would have to undergo any compound dissection routine which was built into the machine; thus all misspelled words and internationals would have to be tested for compounding before they were printed out. Since about ten per cent of ordinary Russian scientific text is composed of internationals, the time consumed in attempts at compound dissection would probably be prohibitive. If only letters at the end of the word were ignored, the ignored portion has not been compared with the memory, and if the routine were halted when all the letters have been used up, then no appreciable skipping about the dictionary would be required. This much of the compound dissection scheme might be profitably used.

There are many possible variations for the order in which words might be stored in the dictionary. They might be stored first according to the length of the word, then alphabetically; the words might be stored first according to grammatical type, then alphabetically; or the storage might be completely alphabetical. Each of these systems has certain advantages, but complete alphabetization has one clear-cut advantage over all other methods ; each word then has a unique address in the store determined by one criterion alone, the spelling of the word. In this paper, only complete alphabetization of the dictionary will be considered.

The permanent memory has two design limits: a maximum allowable access time, and a maximum allowable size. The maximum size of a large permanent memory is mainly determined by physical limitations; for instance, a memory using a photographic plate to store the information may be limited in size by the area of plate which can be scanned. If a particular memory of the photographic-plate type is capable of storing 50 megabits of information, the memory cost will be approximately the same whether it stores 50 megabits or 15. Access time is actually determined by the efficiency of the search routine. Thus it is sometimes better to store more material in order to realize a more efficient search routine. The first criterion for determining the amount of material which must be stored in the memory system is necessarily the linguistic requirements of the problem. Once the linguistic requirements are satisfied, the all-important criterion for determining size of storage, sequence of storage, and the method of coding the material is the effect of each on the access time into the memory. The physical configuration of the memory largely establishes the optimum storage sequence and the most efficient indexing procedure. A likely device for the large memory of the translation machine is an optical device[12] in which the information would be stored on a photographic plate and the plate scanned by a beam of light formed by a cathode-ray spot and focused by a lens. To store the required 40 megabits of information in a memory of this type there might be about 6,500 rows of about 6,500 bits each. This would give a total of 42.5 megabits of storage. The discussion of storage sequence and indexing will be confined to schemes applicable to this type of storage. A hypothetical portion of a photographic plate on which a part of a dictionary *is* stored is illustrated in Fig. 1. Actually the 1's would most likely be just spots on the plate and the 0's just a lack of a spot, but the representation shown is easier to comprehend.

It can be shown that it is important for the largest part of an incoming text word to be correlated first. For instance, if the source language is English, incoming text words should be compared with "manage" before "man," and with "indicated" before "in." For this reason the source language entries shown in Fig. 1 are stored in inverse numerical sequence in a row, i.e., the numerically largest code group in a particular row is stored to the extreme left of the row with its target language entries, then the numerically next largest source language code group, and so on. The entries in column *C* of Fig. 1 are indexing entries which will be explained later. The word codes are entered in the rows starting with column 1, first the source language word code and then its target language equivalents. In Fig. 1, row 3780 stores the code for "офталмоскоп" as its first source language word, and row 3779 stores "официозый" as its first source language word, etc. The word which will serve as an example is "офеня" and it is assumed to be stored in columns 127-153 of row 3778.

If the incoming text word were "офеня," and no indexing scheme were used, the spot would move in a path traced by the solid line in Fig. 1. The circuitry enclosed by the solid line would be used in this case. In each row, in order, the spot would read the code until a binary place was reached which was different from the corresponding binary place in the text-word code. If at that binary place the text code had a "1" and the dictionary code had a "0," the spot would start to read the next row. If the reverse were true, the word must be located in that row if it is stored in the dictionary. The spot would continue reading that row until correlation was established or until a stored source language code sequence had a smaller code number than the text word, which would establish the fact that the text code was not stored in the dictionary. A dissection routine would then be started if such schemes were included in the stored logic. With this search procedure, the spot must read an average of perhaps three or four binary places at the beginning of each row before the machine could determine whether or not it should go to the next row or should continue to read the same row. An indexing scheme may be used to reduce the average number of binary places that are read in a row before the machine would decide whether or not it must go to the next row. This is particularly easy in this example since complete alphabetization of the dictionary is assumed.

The circuitry necessary for one possible indexing scheme is represented in block form in Fig. 1, enclosed in a dashed line. This scheme depends upon the fact that the first two letters of a word will determine closely the row in which the word must be stored. A "1" is placed in column C in a particular row each time the first two letters of the first word stored in that row are different from the first two letters of the first word stored in the previous row. The idea is to treat the first two 5-bit code groups of an incoming text word code as a binary number; the rows which start with "оф" (code 0111010100 which is binary for decimal 468) then form row group 468. For example, suppose that the incoming text word is "офеня" (peddler). The first two letters, "оф" will have the code 0111010100.

This code sequence is transferred to the reversible binary counter, leaving the contents of the register unchanged, The spot then moves down column *C*. subtracting a binary "1" (0000000001) from the counter each time the spot passes a "1" in column *C*. When the content of the counter is reduced to zero, the machine starts reading the rows, as indicated by the dashed line in Fig. 1, exactly in the same way as when no indexing scheme was used. Two complications arise in the scheme: first, the first row of the dictionary starts with 0000100100 (it is assumed that агенц (lamb) is stored in this position), which is considerably larger than 000000001 and, second, some letter sequences do not exist as the first two letters of a word, for instance, there are no words beginning оу . . . . The first complication is resolved by the use of a subtracter between the input register and the counter which automatically subtracts a binary 0000100011 from the 10-bit sequence as it is transferred to the counter. If a negative difference is obtained when this subtraction is performed, the word must be located in the first row. If the difference is positive, the spot moves down column *C* subtracting 0000000001 from the contents of the counter each time it passes a binary "1." The second complication is resolved by placing dummy "1's" in column *C* in rows which have the same first two letters as the preceding row. For instance, alphabetically (and numerically for the corresponding codes) the sequences are "от, оу, оф," but no "оу . . ." words exist. Therefore, to keep the count straight a dummy "1" is placed in column 1, row 3777. The search for the word "офеня" would then follow the path shown by the dashed line. This path would require reading only one binary place per row until the spot had approached within an average of six or seven rows of the desired row.

About 1,200 words[13] constitute approximately 40 per cent of text material in most languages so it would seem profitable to include these words of short length and very high probability of occurrence in a separate section of the memory. Then incoming text words could be compared with this small dictionary before comparison with the large memory. This auxiliary small dictionary would be used if the other machine operations take substantially less time than the dictionary search, since its use would undoubtedly speed up the average location time of words. If the dictionary search was fast enough so that the other machine operations were the limiting factors, then the logical complications of two single word dictionaries would not justify splitting the dictionary.

Idioms could be stored in a separate dictionary and incoming words compared with this dictionary before comparisons with the single-entry dictionary. These idioms might also be stored in the same dictionary as the single-word entries. Storing both the idioms and single-word entries in the same dictionary would probably give a lower average location time, but the logical problems would be more complex. The operation of a separate idiomatic dictionary will not be discussed in detail since source language text material could be correlated with this dictionary in the same manner as with the individual-word dictionary although the routine would be somewhat more complicated. This idiomatic dictionary should be small compared with the individual word dictionary.

To store 40 megabits of information on a photographic place in the manner shown in Fig. 1, a square of information about 6,500 bits on a side is required, which yields a total storage of about 4.25 X10" bits. With 6,500 rows in the large dictionary, the average number of rows passed by the spot before it gets to the proper section is about 3,250. Since the rows also have about 6,500 bits stored in them, the average number of bits read in the proper row would be about 3,250, or a total of about 6,500 bit positions to read, on the average, before a word is located in the large dictionary. If a repetition rate of 100 kc is assumed (this will allow relatively simple transistor circuitry to be used) the average location time for an incoming text word would be about 0.065 second. Thus it seems possible to construct an optical store which would have the required capacity and allow an access time of less than 1/10 second, which was previously estimated as an economical speed of operation.

LOGICAL OPERATIONS

Word-for-word translation has proved surprisingly good, but not satisfactory for commercial translation. Some consideration of the context of words seems necessary to obtain a satisfactory output. Yngve[14] suggests consideration of perhaps three or four words on each side of a particular word when attempting to determine a unique meaning from a set of multiple meanings of the word. He does not suggest the consideration of words not located in the same sentence as the word being processed. To aid in such logical operations, linguistic "tags,"[13] or code sequences which denote certain grammatical information about the word, may be stored in the memory along with the target language equivalents of the word. Every source language word having more than one target language equivalents would require the tag. At present, it is not possible to estimate how much processing would be required, and how much information and of what type should be stored in these tags since linguistic research has not progressed very far in this direction. Some work has been accomplished[13] on grammatical incident meaning but nothing has been done on the nongrammatical incident meaning of words. The logical operations are probably the most important area for contemporary research. To illustrate how the linguistic tags might be used to eliminate impossible target-language equivalents, the following example given by Reifler (page 152 of reference 13) is reproduced. Consider the German expression "wegen dieser Schueler" (because of these pupils). "Dieser" has in isolation the possible grammatical meanings of singular masculine nominative, singular feminine genitive or dative, or plural genitive. The word "Schueler" has in isolation the grammatical meanings of singular masculine nominative, dative, or accusative, or plural nominative, genitive or accusative. Consider the following tag assignments.

1. Singular masculine nominative.
2. Singular feminine genitive.
3. Singular feminine dative.
4. Plural genitive.
5. Singular masculine dative.
6. Singular masculine accusative.
7. Plural nominative.
8. Plural accusative.
9. Singular genitive.

The tags stored with "wegen" will be 4, 9; with "dieser," 1, 2, 3, 4; and with "Schueler," 1, 4, 5, 6, 7, 8. The co-occurrence of "wegen," "dieser," and "Schueler" narrows down the two alternatives of "wegen," the four alternatives of "dieser," and the six alternatives of "Schueler" to only one possibility, that plural genitive since only tag 4 is common to all three.

DECODING

In the decoding operation, material in machine code would be translated in some way into the target-language alphabet. A device has been recently reported which accepts code and translates it into printed form on a cathode-ray screen at the rate of several thousand letters per second.[5] The. material can then be photographed, and erased, and new material can be introduced. This printing speed is an order of magnitude greater than the minimum that is required for a translator.

Conclusions

There appear to be no insurmountable engineering problems in the realization of a mechanical translation machine. Research has progressed sufficiently to allow the formulation of a detailed program which should result in a commercially practical machine in a reasonable time. Much work remains to be done, most of which is linguistic in nature, but a substantial number of good engineering problems remain to be solved.

(*Discussion follows*.)

References

1. MACHINE TRANSLATION OF LANGUAGE (book), A. Donald Booth, William N. Locke, editors. "Historical Introduction," Booth and Locke. John Wiley & Sons, Inc., New York, N. Y., 1955, pp. 2 ff.
2. *Mechanical Translation* (periodical), Massachusetts Institute of Technology, Cambridge, Mass.
3. Booth and Locke, op. fit.
4. COMPUTING MACHINES FOR LANGUAGE TRANSLATION, Thomas M. Stout. *The Trend in Engineering*, University of Washington Press, Seattle, Wash., vol. 1, no. 3, Dec. 1954, pp. 41-46.
5. THE MECHANICAL TRANSLATION OF RUSSIAN: A PRELIMINARY REPORT, Kenneth B. Harper. University of California, Los Angeles, Calif.
6. RUSSIAN-ENGLISH DICTIONARY, Louis Segal. G. B. Stechert and Company, New York, N. Y., 1942.
7. PHOTOELECTRIC READER FEEDS BUSINESS MACHINES, David H. Shepard, Clyde C. Heasly, Jr. Electronics, New York, N. Y., vol. 28, no 5, May 1955, p. 134.
8. Booth and Locke, op. cit. "Some Problems of the Word," William E. Bull, Charles Africa, Daniel Teichroew.
9. ENZYKLOPÄDISCHES ENGLISCH-DEUTSCHES UND DEUTSCH-ENGLISCHES WÖRTERBUCH, Muret-Saunders. Langenscheidtsche Verlagsbuchhandlung. G. m. b. H. Berlin-Schöneberg, Germany, 1910.
10. Booth and Locke, *op. cit*. "The Design of an Automatic Russian-English Dictionary," Anthony G. Oettinger.
11. MECHANICAL DETERMINATION OF THE CONSTITUENTS OF GERMAN COMPOUNDS, Erwin Reifler. *Mechanical Translation*, Massachusetts Institute of Technology, Cambridge, Mass., vol. 2, no. 1, 1955, p.3.
12. PHOTOGRAPHIC TECHNIQUES FOR INFORMATION STORAGE, Gilbert W. King, G. W. Brown, L. N. Ridenour. Proceedings, Institute of Radio Engineers, New York, N. Y., vol. 41, no. 10, Oct. 1953, p. 134.
13. Booth and Locke, *op. cit*. "The Mechanical Determination of Meaning," Erwin Reifler. pp. 159 ff..
14. Booth and Locke, *op. cit*. "Syntax and the Problem of Multiple Meaning," Victor H. Yngve.
15. AN ELECTRON TUBE FOR HIGH SPEED TELEPRINTING, W. H. Bliss, J. E. Ruedy. *RCA Review*, Princeton, N. J., vol. XVI, no. 1, Mar. 1955, p. 5.

Discussion

M. M. Astrahan (International Business Machines Corporation, San Jose, Calif.): The 42.5-megabit storage unit proposed would be impractical with presently available or contemplated components. As pointed out in reference 12 of the paper, a cathode-ray tube with a resolution of at least 6,500 lines would be required. Also the problems of accuracy and stability in addressing would be prohibitive.

I would think that the use of an indexing system could considerably reduce the amount of sequential scanning required so that memories with smaller sized high-speed sequential access sections, and slower access to each of the sections, could be employed.

Robert E. Wall, Jr.: Mr. Astrahan is perfectly correct in stating that an indexing scheme may be used to allow rapid processing of material with a memory system, as he describes. Such memories are available now, while the memories of the type indicated in this paper are not now actually available. Until very recently, a Cievite-Brush TapeDRUM was considered for dictionary storage and was to be used as follows: Incoming text material was to be stored in a temporary memory in blocks of about 1,000 words, then each word was to be indexed according to its alphabetical order. The dictionary would then have been searched in alphabetical order for the entire 1,000 words at one time. It is possible to obtain a processing time of ten words per second with such a scheme. The disadvantage of this method is that considerable temporary storage space is required to store and index the incoming material; considerably more than would be necessary just for the logical processing alone. It can also be shown that any type of word dissection scheme is rendered very difficult with a dictionary stored on a low random-access memory. Any processing in the source language, such as perhaps would be required for processing the negations of French, is also much more difficult to perform.