IBM Research

A TABLE-LOOKUP MACHINE FOR PROCESSING OF NATURAL LANGUAGES

J. L. CRAFT
E. H. GOLDMAN
W. B. STROHM

Revised: February 27, 1961

A TABLE-LOOKUP MACHINE FOR PROCESSING OF NATURAL LANGUAGES

by

J. L. Craft, E. H. Goldman, and W. B. Strohm

ABSTRACT: A table-lookup machine based upon the photographic memory invented by King is being optimized for the processing of natural languages. It makes use of automatic retrieval of lexical information by means of novel addressing features which allow lookup of phrases regardless of length. Linguistic determinations made on the basis of the lexical information retrieved govern subsequent operations. In addition a sentence buffer holds a sentence long enough to make the backward and forward passes which will be required to make grammatical and contextual analyses.

Research Report
RC-307
July 5, 1960

I. INTRODUCTION

For the past decade data processing machinery has been devoted to aiding mathematicians, physical scientists and accountants. The problems to be solved have been rigorously defined by precise symbols and numbers. The most important operations have been characteristically arithmetic. However, the preponderance of human thought must still be conveyed in the relatively inexact words and sentences of natural languages. This has resulted in an increased interest among the computer engineers at the International Business Machines Corporation in developing a machine for processing languages. The principal objective of such a machine would be the extraction of proper meaning; the most important operations are table-lookup in nature.

Automatic translation of languages has become important not only for itself but also as a first step in the development of methods for utilizing machines to assist in human thought. The essential problem is the establishment of equivalence between the signs (i.e., words, phrases, sentences) of two different languages. These signs often have more than one meaning. Resolution of a word with multiple-meaning can be accomplished, at least in part, by a sequence of basic operations and, therefore, comes within the scope of electronic computers. This paper will describe a U. S. Air Force experimental language-processing machine organized to take maximum advantage of table-lookup operations, and based technologically upon the photographic memory invented by King[1, 2] at the International Telemeter Corporation. The machine has been developed under contract with the Intelligence Laboratory of the Rome Air Development Center.

II. THE NATURE OF THE PROBLEM

The three major steps in automatic translation consist of lexical recognition (the dictionary problem), syntax (the grammar problem) and semantics (the meaning problem). These cannot be treated independently but can be analyzed sequentially and repetitively in a number of operational loops. Thus lexical recognition first provides a number of alternate meaning classifications and parts of speech (syntactic categories). Syntactic analysis of inflectional endings and word order then reduces the number of possibilities. In French, for example, most good dictionaries give four possible meanings of "son" as a pronoun (his, her, its, one's) and two possibilities as a noun (sound, bran). That "son" is a noun rather than a pronoun can easily be determined if an article such as "le" or "un" precedes the word. Semantic analysis then must attempt to determine which is the most likely of the meanings which are possible as a noun. Statistically, the other words in the sentence must be analyzed to determine whether they fall in a class of meanings having to do with "bran," such as "cereal," "flour," or "meal," more or less often than meanings having to do with "sound," such as "hearing," "noise," or "communication." More detailed examples of these three steps will be given in the sections which follow.

A. Lexical Recognition

The mechanized translation scheme which has been briefly described above is a practical one only if the electronic system includes a rapid access storage with capacity greatly in excess of customary mathematical computers. Until recently such a memory has not been available, and early research in mechanical translation was limited to the preparation of methods which permit a more restricted memory[3].

In recent years technical advances have made large, fast, random access memories a reality. Magnetic Disc memories with modular units of 5-million bit capacity are already in common usage. The photographic disc memory used in the system described in this article has a 30-million bit capacity with an average random access of 35 milliseconds. Memories with these characteristics make possible for the first time a really useful automatic lexicon of 400,000 entries.

A number of research groups in the United States have been engaged in developing techniques for compiling automatic dictionaries. These include the University of California at Los Angeles[4], Harvard University[5], and IBM Research. All of these dictionaries separate the stems of regular nouns, adjectives, and verbs from the endings and thus save on both memory space and the labor of listing full declensions and conjugations. In French "haute" would not be in the dictionary. It would be found by first locating "haut," then the feminine ending "-e." The IBM lexicon includes idioms and phrases in a system which automatically matches on the longest possible entry. Thus, "haute pression" is recognized before "haut," so that the translation will be the unambiguous "high pressure" rather than a list of multiple alternatives for "haute" (high, deep, upper, bright, loud, etc.). As will be shown later in this paper, the logic for these lexical searches is designed into the addressing system of the memory itself. No programming is required.

B. Syntactic Analysis

The basic problem in syntactic analysis is to provide procedures by which structural patterns in a language can be recognized. The objectives of this analysis are to (1) place the translated words in a proper order and a correct grammatical form, and (2) assist in the resolution of multiple meaning by determination of the correct word category. Georgetown University[6], U.C.L.A.[7], and MIT[8] conducted early research along these lines, while several other research groups, in particular the RAND Corporation[9], are now making syntactic investigations.

The basic operation in syntactic analysis is that of parsing -- the classification of words of a sentence into possible parts of speech, and then the elimination of invalid possibilities by applying rules of syntax. The principal clues at the service of the analytical system are (1) the word root or stem, (2) the inflectional endings, and (3) word order. In some languages, such as Russian, the inflectional endings are extremely useful; in others, such as French and English, word order has a more important role.

An example of syntactic analysis in French using word order is "La theorie des groupes" (the theory of groups). Prior to the application of the rules of word order, if the entire phrase is not entered in the dictionary, the system might recognize "la" as a pronoun (it) or an article (the), "theorie" as a noun (theory),"des" as a partitive article (untranslated) or as a preposition (of), and "groupes" as a plural form of a noun (groups) or the second person singular form of a verb (group). For the sake of illustration, let us now oversimplify some rules of word order to state that: when a word which can be a pronoun or an article precedes a noun, it is an article; when a word which can be a noun or a verb follows a word which can be a partitive article or a preposition, it is a noun; and finally when a word which can be a partitive article or a preposition falls between two nouns, it is a preposition. If these rules are correct and have no exceptions, the ambiguity in syntactic category and also in meaning can be completely resolved to give "The theory of groups" rather than "it theory group." The big difficulty in automatic translation is that grammatical rules are never this simple. Discontinuous constructions (words which are separated in the sentence from modifiers or objects) are particularly difficult to analyze by any known fixed set of grammatical rules.

C. Semantic Analysis

The problem of resolving multiple meanings which are beyond the scope of the large dictionary and the syntactic procedures is probably the most difficult to mechanize. Early work resorted to the printing out of all possible meanings and required a post-editor familiar with the language or subject matter to select the proper meaning. Thus the real advantages of speed and accuracy offered by an electronic system were not realized.

The semantic problem is similar to the syntactic in that it can be resolved only by classifying words into categories. Semantic categories are classes of meaning. They are usually multiple for any word until the number is reduced by means of context. An example of semantic categorization is the Roget type of classification, which has 1,000 categories of meaning. Each word in a sentence can have several possible categories. These categories must be compared with each other, and, since electronic machines can only follow rigorous procedures, the comparison can only be made according to a definite set of rules.

A simplified example of semantic analysis using Roget's Thesaurus can be given by attempting to translate the English phrase "radio broadcast" into another language. The first word belongs to a single meaning group (Roget category 534, concerned with the communication of messages), but the second has multiple meanings (73 and 291 concerned with dispersal, and 531 and 532 concerned with communication). Since the number 534 representing the semantic category of "radio" is very close to 532 and far from 73 or 291, the meaning concerned with communication is chosen over that concerned with dispersal.

In summary, the first requirement of automatic translation is that of a very large, high-speed, random-access dictionary containing idioms and phrases, and preferably an addressing system which selects the longest possible match. Understandable translations have already been produced at IBM Research using such a dictionary. The other two requirements are for rigorous systems of syntactic and semantic analysis using text which precedes and follows the word or phrase under analysis in order to determine proper sentence structure and to resolve multiple meanings.

III. General Machine Description

A simplified pictorial block diagram of the translating system is shown in Figure 1. The heart of this system is the rotating glass disc on which are recorded coded Russian idioms, phrases and words paired with English meanings.* The disc thus acts as an automatic dictionary which is read by means of a cathode-ray tube light source, a moving lens, a photo multiplier tube and some electronic circuitry. In addition to the disc equipment, key portions of the system include the input typewriter which is used for typing the input Russian text, the input register which holds input text until the disc is searched for the proper dictionary entry, the lexical buffer memory, and the output units.

Continuing to examine Figure 1, we can explain the operation of the translation system by tracing the events which flow from left to right when input text is typed on the input typewriter. The input Russian characters are each coded in the form of holes in an input tape. After the tape passes through the tape reader the information, coded into "ones" and "zeros," six per character, is placed in the input register. These characters are then compared with the information being read out of the dictionary in order to determine the proper direction to move the lens and cathode-ray tube beam. This is analogous to a human looking into a printed dictionary at whatever page is open and then flipping pages in the proper direction as a result of reading the first entry. Instead of pages in a printed dictionary we have concentric tracks on the disc dictionary. The light beam continues to step across acks sampling a small portion of each, until the comparator indicates that it has gone too far. The beam is then brought to rest and the disc rotation (1400 r.p.m.) allows the reading of every entry on that particular track. This corresponds approximately with our reading the entries on a page of a printed dictionary from the bottom in order to get the longest possible match (for example, "time constant"

*See Reference 2 for details of the disc technology. A more recent and complete description is given in "Final Report on Computer Set AN GSQ-16 (XW-1)," prepared for Intelligence Laboratory, Rome Air Development Center (RADC-TR 59-110)

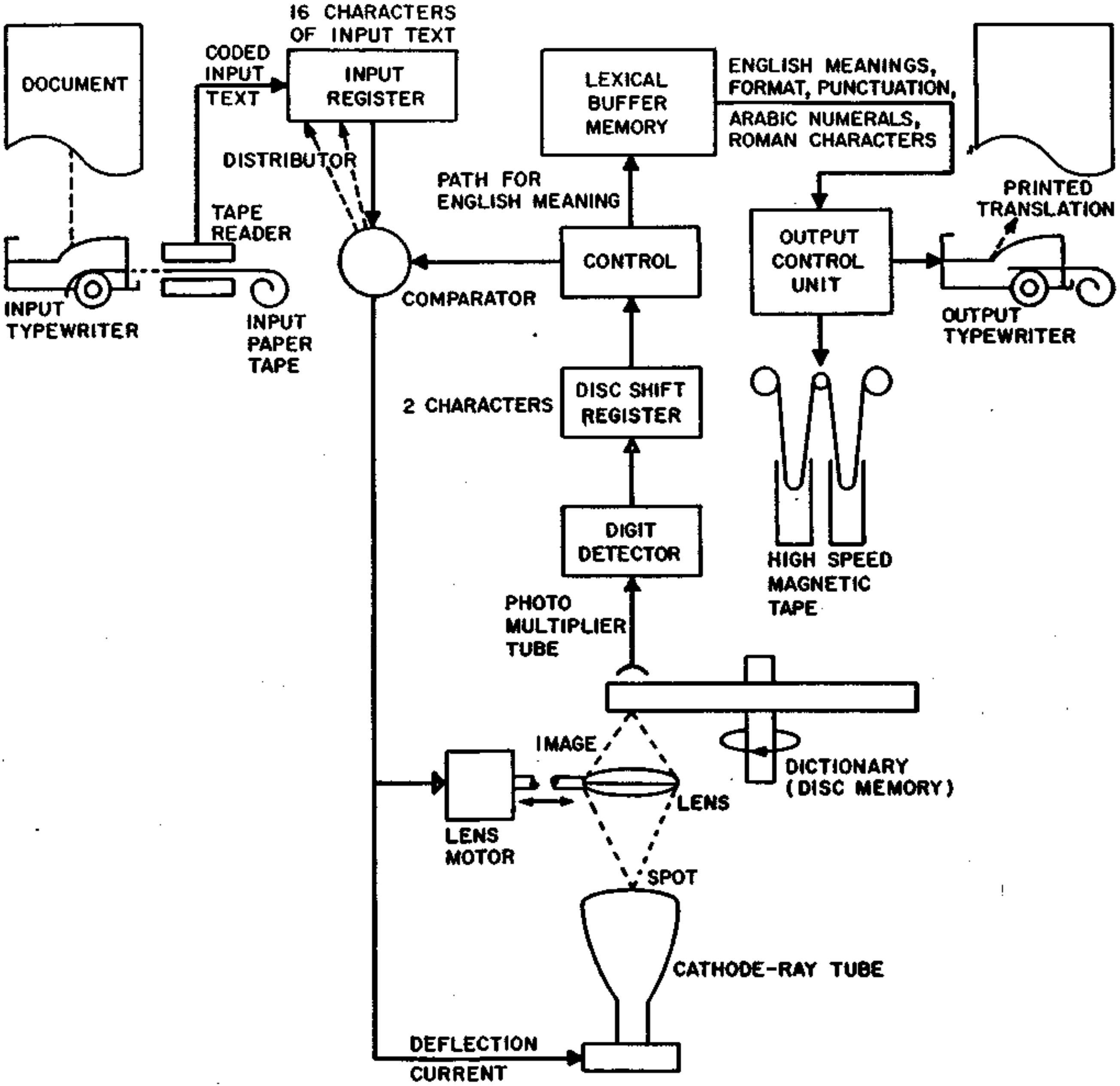


Figure 1 - Simplified system diagram

before "time"). When a proper match to a Russian semantic unit has been found, the corresponding English meaning is read out through the high speed register to the Lexical Buffer Memory. At the same time logical circuitry indicated by the "distributor" has kept an account of the number of input characters for which a match has been found. This allows the input characters which have been used to be discarded and fresh input characters to be shifted into the input register. Address modification in subsequent searches is possible, and will be described in Section IV.

Lexicon entries on the disc are laid out in such a way that the Russian words and idioms themselves make up the address of the entry. Each character in the Russian word has a certain binary code which can be interpreted as a weight. Cyrillic "e" has the lowest weight and Cyrillic "B" has the highest. Each coded Russian word, therefore, looks like a long binary number. The layout on the disc is in numerical order. As the disc is scanned track by track, each bit (one or zero) in the Russian word is compared with the corresponding bit in the input register (which here is acting like a memory address register). This comparison is continued until disagreement is found. At this time a condition consisting of "zero" on the disc and "one" in the input register means "go ahead" to the next track. The inverse combination means "go back." Only when the location of the correct entry has been passed is a particular track scanned exhaustively at a rate of one microsecond per bit. The exact match has been found when each one and each zero in theRussian dictionary entry matches exactly with each one and zero in the input register, until the symbol signifying the end of the Russian word in the dictionary entry is reached.

The Lexical Buffer Memory (Figure 1) is a gathering place for all characters read from the disc until a sentence has been collected; it is therefore sometimes referred to as a "sentence buffer." The sentence is transferred as a block to an output unit as shown, or it can be held for analysis and recirculated to the disc memory to gain further information. The linguistic theory for sentence analysis is still being developed.

The cathode-ray tube shown in Figure 1 is used because an electron beam can be moved faster than any other light source. When it is necessary to move the light from one disc track to another, the change is accomplished rapidly by means of the deflection current. The lens motor, with its higher inertia, moves more slowly and allows the cathode-ray tube electron beam to return to the center of the tube face. Thus the rapid movement of the electron beam makes possible the low access time (35 milliseconds average) while the lens motor prevents the electron beam from going too far toward the side of the cathode-ray tube.

The design of the dictionary has utilized photographic techniques because photographic emulsions are the densest storage media known today. Although this storage is permanent, a great deal of work has gone into the design of equipment which can prepare new discs rapidly, thus allowing frequent updating of the list of entries in the dictionary.

Except for input-output equipment, the speed of the system in Figure 1 is sufficient to process Russian technical literature at approximately 30 words per second. Only the relatively slow speeds of the input typing and output printing limit the speed of the overall system to a lower rate. In the future it is expected that the input speed limitation will be removed by the use of automatic page scanners and character sensing. The output speed limitation can be eliminated by means of high-speed printers and by multiplexing several output units.

IV ORGANIZATION OF THE LEXICAL FILE

The lexical information recorded on the disc is arranged serially by bit and character along the various tracks. A lexical entry consists of an address containing the characters of the object language, an output containing the characters of the meaning in the target language, and several control instructions. Since words and phrases of natural languages are variable in length, the disc entries vary similarly. The control characters serve the function of identifying entries and separating the address from the meaning. A dictionary entry can be represented as follows:

$$- - - \; M\,M\,M\,\underbrace{\alpha\,A\,A\,A\,A\,\tau\,M\,M\,M\,M}_{}\,\alpha\,A\,A \; - - -$$

The instruction "α" identifies the beginning of entry, synchronizes the search system and institutes comparison between input data and the characters "A" of the entry address. The instruction "τ" serves to terminate the address and, during a file search, indicates a match with corresponding input characters, initiating the read out of the characters "M" of the meaning to the Lexical Buffer. The "α" of the next entry serves to identify the end of the meaning, terminate the read out, and start a shift operation in the input registers to shift in new information and to discard the information just matched.

The entry address and meaning are separate and distinct. The organization of the file imposes no restrictions on the contents of either. It is possible to code each differently, thereby obtaining a code translation function. In practice the addresses may contain characters of the object language, punctuation or numerals. The meaning or output may correspondingly contain characters of the target language, punctuation or numerals. Also included are certain edit symbols which control textual format.

A. Integral Addressing

The lexical entries are ordered on the disc in descending numerical sequence as determined by the binary value of their address. This binary value depends not only upon the code value of the individual characters in the address but also upon the address length. As in a conventional dictionary, the first character has the highest weight: a longer address, such as "manipulate," has a higher value than the address "man." Thus a hypothetical sequence of addresses might be "--- men ----manipulate ---- man ---."

A track search of the lexical file is first directed in such a way as to locate a track which contains disc entries of higher binary numerical value than the input data. A sequential, entry-by-entry search follows in the direction of lower-valued entries. The first matching disc entry is then the longest possible match with the input data.

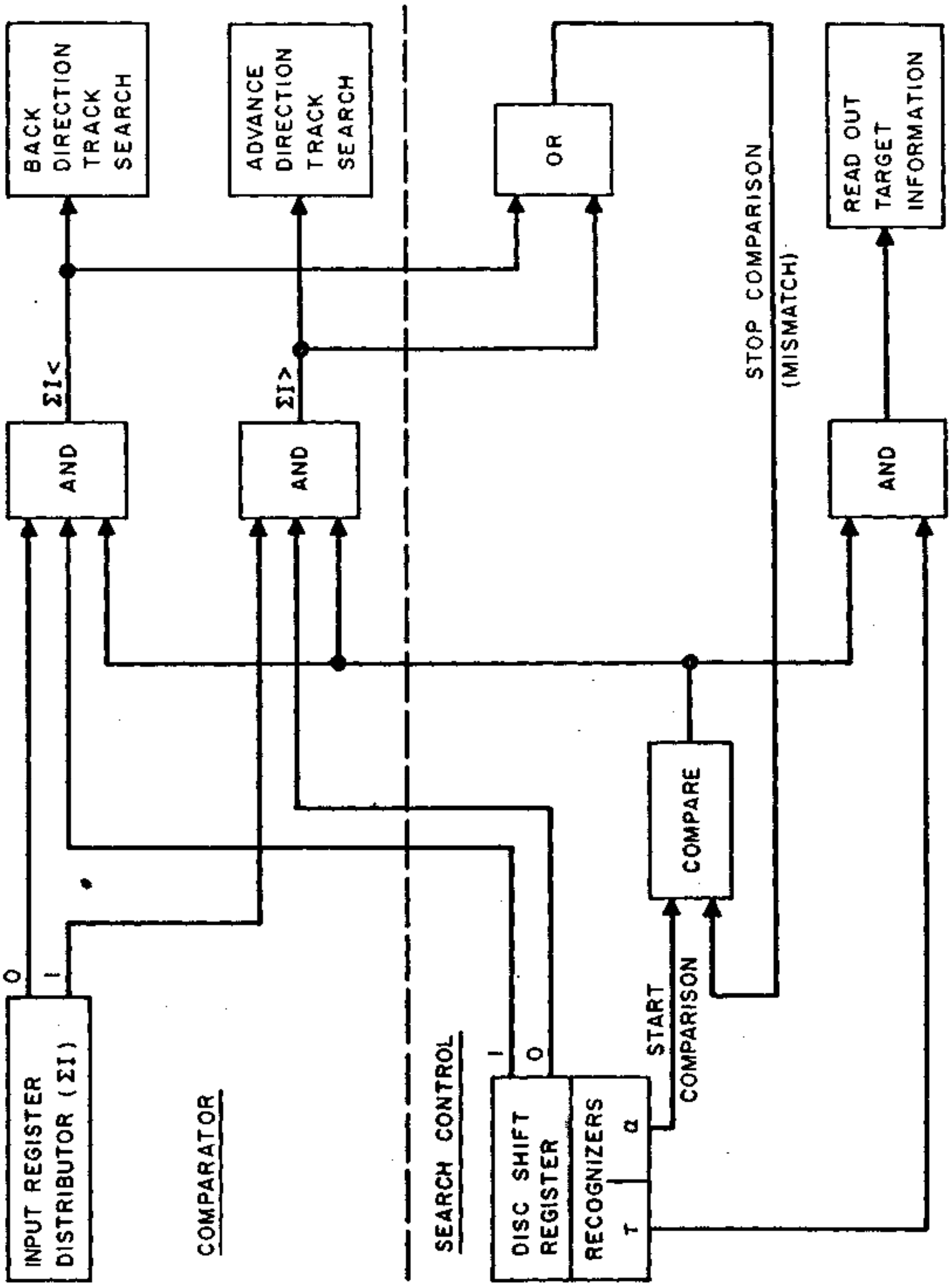
The validity of this lexicon search algorithm becomes apparent with a more detailed analysis of the lexicon ordering and the search operations. Entries are arranged consecutively on a track which begins with the higher-valued addresses and ends with the lower-valued addresses. Tracks are located concentrically and are arranged in an increasing binary order towards the outermost track. Thus a complete scan of the lexicon in a direction of decreasingly valued addresses would commence with the beginning of the outermost track and terminate with the end of the innermost track; at the end of each track a jump to the beginning of the next lower order track would continue the scan in the desired sequence.

The lexicon search consists of an ordered sequence of comparison routines wherein disc entries are compared to the input data. During each comparison the sequential bits in the stream of disc information are compared bit for bit with the numerical value, $\sum I$, of the input. The numerical value, $\sum I$, is obtained from the distributor, shown in Figure 1, which sequentially scans all the binary information held in the Input Register in synchronism with the disc information entering the comparator.

To search the lexicon for a match with the input data, a track sampling search is first instituted to locate the lowest ordered track wherein a sampled entry is found whose numerical value is higher than the value, ΣI, of the input data, regardless of length. On this track an entry search is begun wherein each lower-valued disc entry is compared bit for bit with ΣI. If a match is not found before the end of this track is reached, a back track to the beginning of the next lower ordered track is effected and the entry by entry comparison is continued. This process is continued until a matching entry is located. During a track sampling search the location changes are effected by track stepping, whereas during the detailed entry search, location changes are effected by disc rotation coupled with occasional back track steps.

An illustration of the comparison technique and related control operations is shown in the logic block diagram of Figure 2. Comparison starts when an "α" instruction is recognized in the disc shift register. The input distributor is simultaneously synchronized. Thereafter the sequential bits of disc information and of the input value, ΣI, obtained from the distributor, are compared in two circuits for inequality. These circuits may indicate either $\Sigma I <$ or $\Sigma I >$ depending on the type of inequality existing in the information. The former indicates that the value ΣI is less than the disc entry address being read from the lexicon. These two comparator signals control the direction of the track search and result in track stepping in the lexicon in the direction of convergence on a matching entry. These comparator signals also stop comparison for the remainder of the entry causing the mismatch. The next "α" instruction recognized causes the comparison to continue. When an entry is reached wherein no mismatch occurs up to the "τ" in that entry, then another recognizer initiates the read out of the target information.

When a track sampling search is initiated, the lexicon entry being scanned at that instant is compared to ΣI. The results might indicate that ΣI is less than, i.e., $\Sigma I <$,


INPUT REGISTER DISTRIBUTOR (ΣI)
O
I
COMPARATOR
SEARCH CONTROL
DISC SHIFT REGISTER
I
O
RECOGNIZERS
T
α
START COMPARISON
COMPARE
STOP COMPARISON (MISMATCH)
AND
AND
AND
ΣI<
ΣI>
OR
BACK DIRECTION TRACK SEARCH
ADVANCE DIRECTION TRACK SEARCH
READ OUT TARGET INFORMATION

the disc entry, and back tracking is necessary,or it may indicate that ΣI is greater than, i.e., $\Sigma I >$, the disc entry, and advance tracking is necessary. An illustration of both of these eventualities is given in Figure 3.

For the case where the initial track sample yields $\Sigma I >$, an advance track sequence commences, wherein the next track is sampled. If this sample produces another $\Sigma I >$, the process continues. The track sampling ends when a $\Sigma I <$ comparison occurs. At this point the detailed entry search, previously mentioned, is made. The entry search is thus started only after the track sampling has passed over, in an advance direction, the track location wherein the matching entry would be located if it is present in the lexicon. The appositional case where the initial track sample yields $\Sigma I <$ causes a back track sequence which continues until the first track is reached which produces a $\Sigma I >$ signal; at this point the track stepping reverses to the advance direction and continues in a manner identical to the first case.

This sequence of search operations guarantees that the longest possible match will be made with the input data. As can be seen from the illustrations, this feature is not only automatic and inherent in the search routine outlined, but is also very close to the shortest converging routine possible without making use of complicated track indices.

The usefulness of the integral addressing feature is apparent in the selection of a proper match for an input word such as "manipulate" where a shorter match on the entry "man" would be improper. Of equal value is the ability afforded to match automatically on word groups or idiomatic phrases, since many linguistic ambiguities can be resolved by the contiguous text. This same feature permits the separate storage of word stems and endings, as in the processing of a highly inflected language where storage of all the inflectional forms is inefficient. In this case the longest match possible would be with the word stem since the complete form is not stored. It is also possible to match on certain punctuation sets which provide syntactical information for analysis and format control. It is possible to distinguish, by appropriate dictionary entries, groups of punctuations and spaces which indicate the termini of sentences, paragraphs and complete texts.

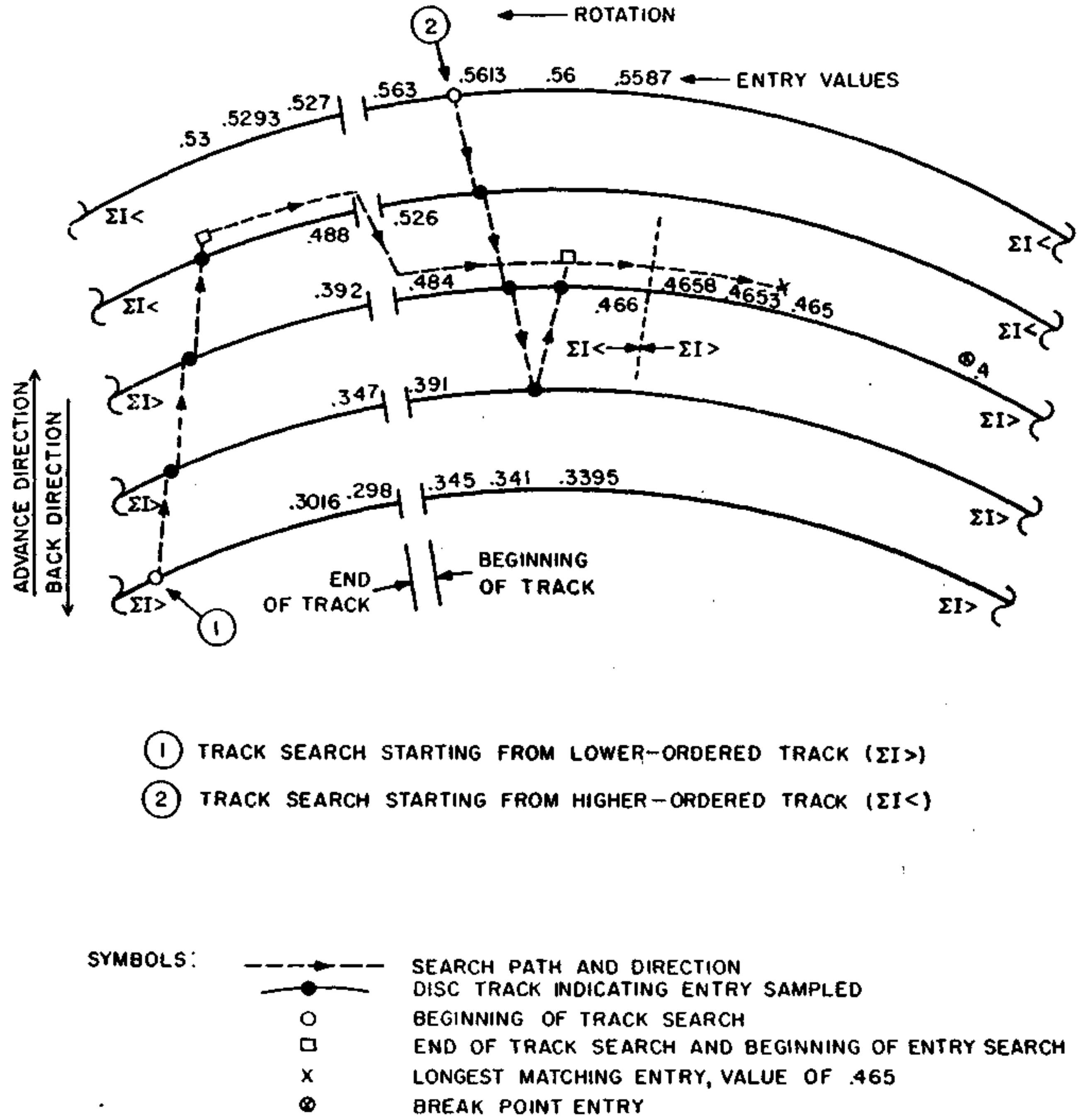

ROTATION
ENTRY VALUES
.53
.5293
.527
.563
.5613
.56
.5587
ΣI<
.488
.526
ΣI<
.392
.484
.4658
.4653
.465
.466
ΣI<
ΣI>
ΣI<
ΣI<
ΣI>
.347
.391
ΣI>
ΣI>
.3016
.298
.345
.341
.3395
ΣI>
ΣI>
ΣI>
END OF TRACK
BEGINNING OF TRACK
ADVANCE DIRECTION
BACK DIRECTION
4
1
2
(1) TRACK SEARCH STARTING FROM LOWER-ORDERED TRACK (ΣI>)
(2) TRACK SEARCH STARTING FROM HIGHER-ORDERED TRACK (ΣI<)
SYMBOLS:
SEARCH PATH AND DIRECTION
DISC TRACK INDICATING ENTRY SAMPLED
BEGINNING OF TRACK SEARCH
END OF TRACK SEARCH AND BEGINNING OF ENTRY SEARCH
x
LONGEST MATCHING ENTRY, VALUE OF .465
BREAK POINT ENTRY

Figure 3 - Example of lexicon search for a match with input value, ΣI, of .46593--

B. Break Points

The search routine previously described must result in some type of match. If the input word being searched in the lexicon is a proper noun, a matching entry for this word will probably not exist. To minimize wasteful search time, break-point entries have been included throughout the lexicon. These entries contain only one or two characters of the object language as their address, there being at least one such entry for each character of the object language. The break-point entries represent the shortest match possible and would match only after the appropriate longer entries have been scanned. The integral addressing automatically results in a search for the longest possible match; if this is a single character break-point entry, then the input word is obviously not stored in the lexicon. Recognition of a break-point entry is employed to start transliteration of the input, where the transliterated "meaning" is the phonetically equivalent target character or characters. The transliteration continues by means of a special addressing routine until the next input word is reached. This routine will be discussed in section C. There are some single character words in most languages which might be confused with break-point entries. For this reason these single character words are stored in a lexicon entry along with the succeeding space or punctuation.

C. Conditional Addressing

A useful and powerful routine has been developed wherein the lexicon entry matching particular input information can influence the selection of the succeeding entry. In this routine, address modification of the succeeding input is achieved by prefixing. Prefix information is stored in a special input register which is scanned as a prefix to the main input register. The control exerted by the first entry requires the use of a control character in that entry. The following are examples:

Disc Entries 1) --- $\alpha A_1 A_2 A_3 \mu \rho_1 \rho_2 \tau$ MMMα ----

2a) --- $\alpha \rho_1 \rho_2 A_4 A_5 A_6 \tau$ MMMα ---

2b) --- $\alpha \rho_1 \rho_2 A_4 A_5 A_6 \mu \rho_3 \tau$ MMMα ---

Input Data $A_1 A_2 A_3 A_4 A_5 A_6$ # ----

The new instruction in entry 1) is "μ" which serves to indicate that a match has been found and that the following characters ρ_1 and ρ_2 are to be prefixed to the next input, A_4 A_5 A_6 # ----. The instruction "τ" serves to terminate the address modifiers and to start the read out of the entry meaning. Again the instruction "α" designates the beginning and end of the entry. Thus the complete address of entries 2a) and 2b) is "ρ_1 ρ_2 A_4 A_5 A_6." When a match is made on an entry such as number 1 above, the modifying characters ρ_1 and ρ_2 are transferred from the disc shift register to the special input registers and circuits are activated which will produce a modified search for the succeeding input after the entry meaning readout is complete and "A_1 A_2 A_3" has been shifted out of the input registers. The next search will be made for the modified input: ρ_1 ρ_2 A_4 A_5 A_6 # ---. Either entry 2a) or 2b) would be in the dictionary; entry 2a) would be employed if the address modification were to be discontinued; however, entry 2b) could be used to continue this routine.

It is significant to note some characteristics of this routine. The modifying character or characters "ρ" are prefixed to the input. In this position they exert utmost control since they occupy the most significant position. In general "ρ" may be selected from the input characters or it may be made unique to initiate a special lexical operation. The address modification routine can be continued indefinitely to include any number of succeeding input characters or words and may be terminated at will by the lexicographer with the appropriate choice of an entry of type 2a).

The utilization of this operation is basic to the present mode of translation in the AN/GSQ-16 (XW-1) system. The most frequent use is found in the splitting of word stems and endings where address modification is employed to associate the separately stored stem and ending. In processing Russian, three unique "ρ" modifiers are employed, one each for noun, adjective and verb endings. The stem matching entry initiates the conditional addressing by prefixing one of the unique "ρ" modifiers. The endings are prefixed and searched as a unique class of data, avoiding improper matches with similar forms. In some cases, where there is syntactic ambiguity in the stems, either partial or complete listing of the inflectional forms is necessary. The splitting of stems and endings, with separately stored entries for each, results in a considerable economy of storage capacity for a highly inflected language such as Russian, but requires two lexicon searches for each complete input word so stored.

Another important use of conditional addressing is in the transliteration of input data not stored in the lexicon. A proper noun probably will not be stored in the lexicon unless it is in common use. The single or double character break point entry will be matched. Now this entry has the form of entry 1) above; that is, it is an entry designed to initiate a conditional addressing routine. The modifying character "ρ_x" used for transliteration is unique and is employed to prefix the remaining input characters of the word being transliterated. After the transliterated meaning is read out from the break point entry, the search for the prefixed remaining characters will result in a match on one of a special set of entries of type 2b), referred to as transliteration continuation entries. These entries have as their address the "ρ_x" modifier and one of the object language characters. The continued use of "ρ_x" for prefixing will constrain the search for the remaining input characters to this set of entries, each one of which produces a transliterated meaning of the single input character addressed.

This sequence is terminated when a prefixed punctuation or space is matched. The terminating entry has the form of 2a) and discontinues the conditional addressing routine. As an aid to the linguist in detecting words not stored in the lexicon, the initiating or break-point entry causes a readout of the "print red" code and the terminating entry reads out the "print black" code, so that the entire transliteration is printed in red.

The same type of routine is employed to transfer, through the system, English words which are entered as titles or explanatory notes. Since the Roman Characters and Arabic Numerals employ, in the input transcriber, the same code as characters of the object language, an input key is employed to indicate the beginning and end of such input sequences. The beginning instruction is matched by an entry of type 1) which initiates the conditional addressing routine and prefixes a unique modifier "ρ_e." Thereafter each succeeding input character is similarly prefixed and the appropriate meaning read out until the ending instruction is matched, terminating the routine.

Another important use of the prefixing feature is the extension of the fixed length input register. The present system is limited, for any one search, to scanning 16 input characters in addition to the prefixed characters. Fór input words of greater length, an artificial dissection of the word is made such that no component disc entry exceeds this limit. This is accomplished by storing an entry for each component and associating these components by address modification. After the last component is matched, the entire meaning is read out. The effect of this is to make the input register appear semi-infinite in length.

Some semantic ambiguities can be resolved through the use of the conditional addressing routine. Where the correct meaning of a word is dependent upon the preceding word, address modification can be employed to associate this pair. The preceding word then would match on an entry of type 1). If the next word is the anticipated one, a matching entry will provide the correct meaning. If the next word is not the anticipated one, then the "ρ" prefix would be dropped when a break-point entry is reached and the next word would be searched without a prefix. This sequence typifies the conditional addressing action. An example of this type of association, where input word one modifies word two, to produce a special meaning for the latter, is given:

Disc Entry	Word One	$\alpha A_1 A_1 A_1 \mu \rho_{1-2} \tau M_1 \alpha$
	Word Two	$\alpha \rho_{1-2} A_2 A_2 A_2 \tau M_{1-2} \alpha$
	or	$\alpha A_2 A_2 A_2 \tau M_2 \alpha$

The modified meaning of word two, M_{1-2}, is obtained only if word one precedes it in the text. Otherwise the second entry for word two would be matched, yielding the regular meaning, M_2.

The address modification inherent in conditional addressing can be employed to control a sequential program type of operation. As shown previously, the modifying prefix can constrain the search to a fixed set of lexicon entries. The initiation and termination of the routine is responsive to specific input information. These features are amenable to multi-level processing operations, with the basic control being exerted by the "ρ" prefix.

D. Special Operational Features

Several special features are included in the search routine to improve overall operation by increasing system flexibility, reliability and continuity.

1. A Character Skip operation provides a means of skipping comparison on any particular input character. The skip instruction "ν" may be included in the address of any disc entry. When the comparison routine reaches this instruction character, comparison is suspended until the next character is reached. Thus if "ν" were the third character in an entry address, no comparison would be made with the third input character. The "ν" character can be used conveniently in entries to skip over some input information, such as an ending, when this data is not essential to the translation. By so doing the multiple listing of inflectional forms is avoided.

2. An Automatic Space feature facilitates the processing of the space character between input words. The space immediately following an input word will be matched along with that word without requiring that the space be included anywhere in the matching disc entry. The disc entry has as an address only the characters of the input word. If a space follows these characters in the input register, this space will be read out with the target meaning part of the disc entry. Were this not the case, then either a separate search would be required for the space or else the space character would have to be included twice in each appropriate lexicon entry.

This automatic space feature is employed only after the search routine has located an entry in the lexicon which matches with certain input information; if this is so, continuance of the comparison determines whether a partial match exists between the disc control instruction "τ" and the next input character. By semi-common coding between the space character and "τ," it is possible to determine whether space immediately follows that input information which matched with the lexicon entry address. When an automatic space match has been made, the target meaning read out is extended to include "α," which is interpreted as space by the output devices. Since the space is treated as a character, it is possible to include it in an entry such as an idiom or phrase.

3. Input Data Shift Control provides a means by which the normal sequence of shifting in new input and discarding matched input may be altered by a suitably programmed instruction in the lexicon entry. An example of this type entry is:

$$- - M\ M\ \underbrace{\alpha\ A\ A\ A\ \tau\ n\ \delta\ M\ M\ M}\ \alpha\ A\ A\ - - - -$$

The two characters "$n\delta$" are added to the target part of the entry when input shift control is desired. The identifiable instruction is "δ" which causes the character "n" to be transferred to the shift control circuits. The ficticious character "n" is selected to have a binary coded value equivalent to the number of input shifts desired after the target read out is complete. The number of shifts programmed may be arbitrarily selected from zero to 16. The instructions "$n\delta$" are not read out to the Lexical Buffer Memory with the other target characters since the present shift control applies only to the input data. An extension of this feature could control the disposition of the output data transferred to the buffer.

This feature can be used to resolve meanings of adjacent words by multiple usage of part or all of the input data. It has been employed to extend the usefulness of transliteration routines and to implement format control.

4. Duplication of Lexicon Entries. Where extreme reliability is required, lexicon entries may be duplicated or repeated many times by means of a special conditional addressing routine. Lexicon duplication may be complete or may be limited to critical control entries such as break points. This can be accomplished without additional controls or circuits. Each entry to be duplicated is stored once in its original form and once with a unique "ρ_D" prefixed. The coding of "ρ_D" is selected to guarantee physical isolation of the duplicate lexicon from the main lexicon, but still permit access by means of the search routines. A search would normally be directed through the main lexicon but would be directed automatically to the duplicate lexicon if a break-point match were to result from this search. If a match still were not found, transliteration would be instituted. The duplicate lexicon technique is employed in conjunction with an error detection device which indicates when erroneous information is read from the photostore. The signal from this detector will cause the search system to stop comparison on an erroneous entry and to ignore any control instructions or "ρ" prefix contained in that entry. Should an error occur in the entry intended to match the input present, then the following break-point match will direct a search of the duplicate lexicon for a match.

The duplicate lexicon technique is just one application of the lexicon division possible with conditional addressing. Conditional addressing can effect a major division of the lexicon into micro-glossaries for reference with special input categories.

V. THE SENTENCE BUFFER

As mentioned early in this paper, the three major steps in language processing consist of lexical recognition (dictionary references), syntax (grammatical analysis), and semantics (contextual analysis). Section IV has described the lexical file and the various techniques for lexical recognition. In order to perform grammatical and contextual analysis it is necessary to include in the system a means for temporary storage of a portion of the lexically recognized text. Since analysis is usually made of words and phrases in relation to other parts of a sentence, the storage unit in this system (titled Lexical Buffer Memory in Figure 1) has been made large enough for full sentences.

The nucleus of the Lexical Buffer Memory system is a two-microsecond coincident-current core array. It provides a perfect speed match for the projected photostore readout rate. It is expected that within the next year, the photostore bit streaming rate will be increased from 1.0 to 3.0 megacycles per second by means of new developments in disc technology. The language processing system utilizes six-bit bytes for the representation of alphanumeric characters, ρ prefixing symbols, and other photostore address modification entries; the resultant byte streaming rate is 500 kilocycles per second. The period of this byte stream is thus 2.0 microseconds, which coincides with the cycle time of the core array. The buffer utilizes coincident-current selection in order to permit reasonable storage size at modest cost and provide excess capacity for expansion as linguistic experience with the system is gained.

The processing of natural languages by table look-up techniques is carried out asynchronously in order that the system may act as soon as lexical information is read out of the photostore. The Lexical Buffer Memory is designed for completely asynchronous operation; storage references may be made at any rate from zero to the maximum 500 kilocycle-per-second rate of the system.

The logic design philosophy that directed the development of the Lexical Buffer Memory will first be discussed considering only output buffering implications. It is to be observed that one of the principal design goals in the synthesis of the language translation facility was the preservation of flexibility of organization, amenable to modification as progress in linguistic research develops. For these reasons storage capacity and the logic capabilities have been assigned expansion factors commensurate with reasonable construction cost.

The organization of the lexical buffer system, when used with an IBM 729 II Magnetic Tape Unit and an output typewriter is depicted in Figure 4. The IBM 729 II records information at either 200 bits per inch per track or 555 bits per inch per track. These recording densities coupled with a tape transport velocity of 75 inches per second produce bit rates of 15 kilobits per second per track or 41.7 kilobits per second per track, respectively. Magnetic tape prepared in the low density mode is compatible with any of the IBM 700, 1400, or 7000 Series computers; magnetic tape prepared in the higher density mode is compatible with the IBM Series 7070 or 7090 data processing equipment. Thus magnetic tape prepared in the low density mode could be analyzed on any IBM medium or large-scale data processor. This feature is invaluable for continuing linguistic research, permitting ready statistical analysis of language parameters.

The typewriter is currently used to produce a human-readable copy of the output information. A paper-tape punch operating in synchronism with the typing mechanism may also be selected if a reproducible record is required. Although the typewriter operating speed is too low for efficient on-line operation, the advantages of typewriter print quality, upper and lower case availability, color shift, and format control are desirable for intermittent monitoring and off-line printing. Color-shift control is used to highlight input words (which are typed in red as a phonetic transliteration) that are missing from the dictionary, or proper nouns which would not be translated by the system. Frequent occurrence of a word typed in red which is not a proper noun would signify that this word should be added to the photostore.

A sentence constitutes a unit record both within the buffer memory and upon the magnetic tape prepared by the system. The sentence is, of course, a variable length unit record. Present processing applications handle the data one sentence at a time. Linguistic experience with the system may indicate that larger blocks of material shoud be processed at a single pass. An average sentence contains 20 words with typical word length of 6 characters. The present coding scheme utilizes 6 binary bits per character; thus the average sentence could be stored by 720 bits of storage. Additional storage must be provided for longer sentences, for tags and other format and record identifying bits. As may be seen from Figure 4, the lexical buffer has a 64 x 64 x 18 core array (73, 728 bits of storage). It appears unlikely that any meaningful object or target language sentence could exceed the capacity of the buffer. However, as syntactic and semantic research advance and iterative processing of data by

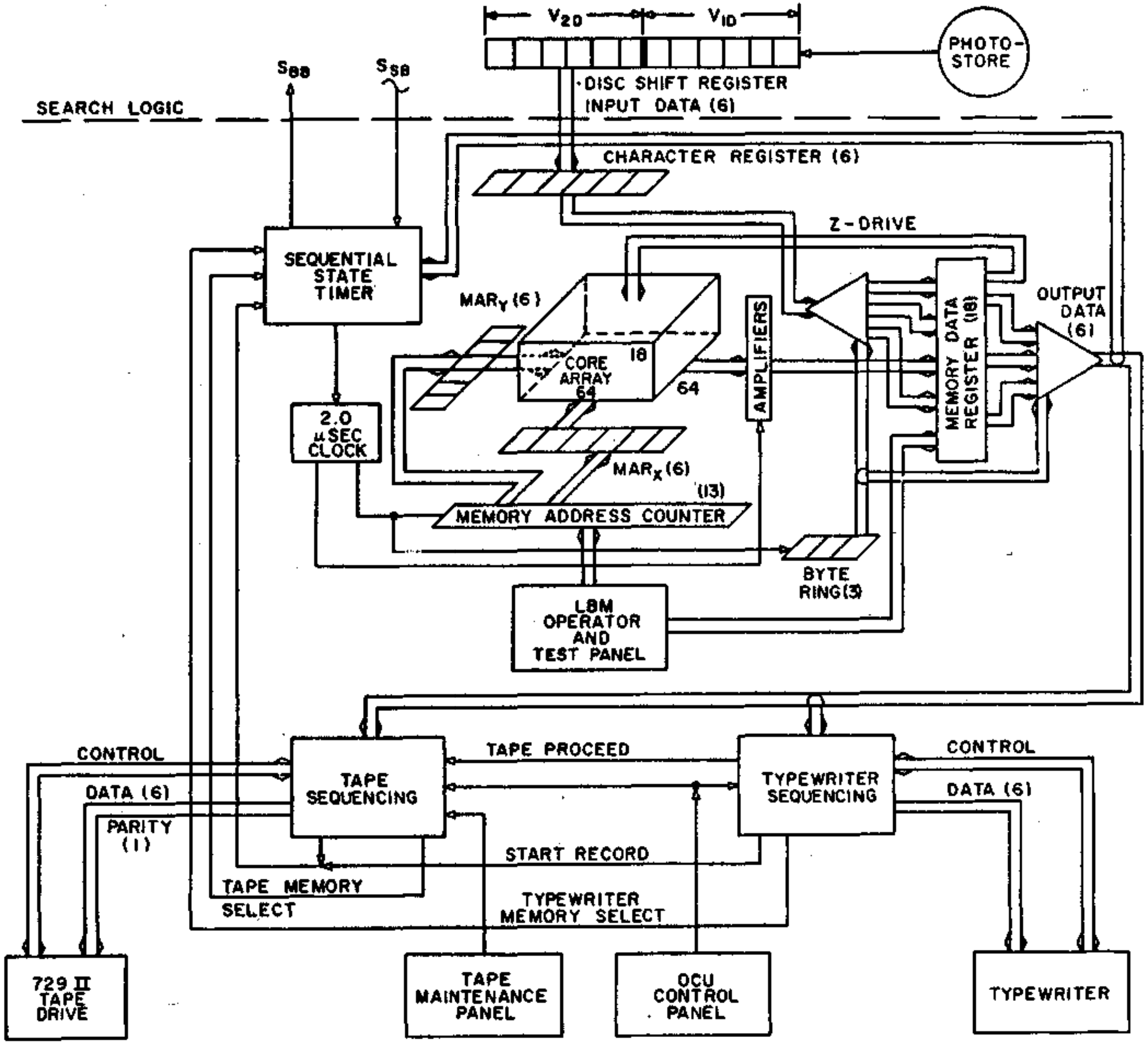


Figure 4 - Lexical buffer system

successive table look-ups is practiced, the "lexical sentence"--which represents an intermediate step between the object language sentence and the target language translation -- may require many more than the average 720 bits previously described. Parsing descriptor tags for syntactic analysis of the sentence or thesaurus descriptor tagging for semantic analysis are representative of the additional binary information which must be manipulated in the handling of the lexical sentence. Partitioning of the memory array for the multiplexing of input and output devices will also use portions of the available memory cells. It will be shown that the system organization of the buffer will accommodate this wide range of eventualities.

Word organization within the buffer is fixed by the following considerations:

a. The magnetic tape unit records information in a parallel-serial fashion. Information is recorded across the width of the tape by six parallel information bits and a parity bit. A record consists of a serial set (along the length of the tape) of these six-bit groups.

b. The output typewriter utilizes parallel six-bit characters as an input, and also records in parallel-serial on its paper tape.

c. The output registers of the AN/GSQ-16 (XW-1) search logic (labeled V_{2D} and V_{1D} in Figure 4) are designed to convert the serial information stream from the photostore into six-bit parallel groups because six-bit coding is presently used on the photostore disc.

From the preceding it seems that six-bit parallel organization is indicated. In addition, it may prove desirable to connect the buffer directly to a data processing system such as the IBM 704 or the IBM 7090 for on-line statistical studies and experimental simulation. A system capable of handling information in certain integral multiples of six-bits is quite adaptable to the 36-bit full word, 18-bit half word organization of the IBM 704 and 7090 computers.

Analysis of engineering considerations and the requirement of the buffer to meet foreseeable lexical research requirements determined the array size to be 64 x 64 x 18. The X and Y dimensions are integral multiples of 16 for engineering reasons; the Z dimension is an integral multiple of 6 for facile input-out communication and an integral sub-multiple of 36 for direct interconnection with data processing equipment.

With this array size, the six-bit parallel consideration encountered during the handling of information for the magnetic tape, the typewriter, and the photostore serializing registers may be implemented by inserting or extracting information into or from the magnetic core array in six-bit "bytes." The processing of the byte-size information is accomplished by extracting the full 18 bits from a memory address. This data is then directed into the conventional memory data register. The proper six output bits from the 18 contained in the memory data register are selected by the three-position byte ring (see Figure 4). Thus to completely specify the address of a byte stored in the buffer memory, it is required to specify the full-word address by means of 12 bits placed in the memory address register and the bit contained in the byte ring. For direct connection of the buffer to a data processor, the byte circuitry is bypassed and the 18-bit information transferred directly.

The signal shown in Figure 4 and labeled S_{SB} (Status-Store Buffer) is used to indicate that a character to be stored is in half of the disc shift register, V_{2D}. This signal initiates a buffer store cycle during which the character is first placed in the character register, then into the proper 6-bit byte position of the 18-bit memory data register and finally into the proper core register position of the array. The triangle preceding the memory data register represents the operation of directing the six input data bits into the proper six-bit positions of the 18-bit memory data register. The byte ring directs the incoming six data bits to the proper byte position of the memory data register and controls the regeneration of the other 12 bits of the 18-bit buffer full word. During insertion of six new bits, the entire 18 bits of the full-word address are subjected to a two-microsecond read-write cycle. All 18 bits are read and cleared to zero during the "read" portion of the cycle; the information from the six bits to be stored is not retained by the memory data register as this information is to be replaced by additional data; the information from the other 12 bits of the full word is retained by the memory data register and is restored into the cores during the "write" portion of the cycle.

The triangle succeeding the memory data register represents the operation of selection of the proper 6 out of 18 bits to be placed upon the output busses of the system. This is also under the control of the byte ring.

A description of buffer operation during a direct object language to target language mode of operation might serve to further clarify the organization of the buffer. At the beginning of the translation cycle, the 13-stage memory address counter and the byte ring are cleared to $(00000)_8$ and $(100)_2$ respectively. The first six-bit target language character is stored in the first six bits of address $(00000)_8$. The next six-bit byte will also be stored at $(00000)_8$ but the byte ring will have advanced so that this byte will be stored in the second 6-bit group of the 00000th full word. The third group will then be stored in the third byte, completing the filling of the 18 bits of the first full word address. The memory address counter is advanced to $(00001)_8$, the byte ring resumes its cycle and loading continues until the end of the unit record (end of target language sentence) is signalled. "End of sentence" is recognized by an end-of-record character.

Because the sentence has been chosen as the unit record, the end-of-sentence or end-of-record criterion chosen for the language translation system is the sequence "punctuation, space, space, capitalize" contained in the input text. This is the normal sentence ending employed in the typing of continuous text. This input sequence is looked-up in the photostore and the punctuation and end-of-record symbol are read from the photostore to the buffer. The next sentence is begun by the read-out of the space-space-capitalize symbols from the photostore. This is the general operation for continuing text; additional considerations are involved for sentences ending a paragraph and sentences ending a complete text or article.

It is to be noted that a counter of only 12 positions is required for the sequential addressing of 64 x 64 full-word locations; the thirteenth position is used to provide a static indication of the"full" status of the buffer and inhibit the storage of further information.

If both present outputs of the system are to be utilized, the operation is summarized by the following sequence: 1) The buffer is loaded with a sentence unit record from the search logic; 2) this sentence unit record is unloaded one character at a time to the typewriter; 3) the same sentence unit record is unloaded to the magnetic tape; and 4) the buffer awaits further loading instructions from the search logic.

During the buffer-to-output-unit transfer of the sentence unit record, the search logic looks up the first word of the next sentence of the object language text. If the search logic finds this first word before the sentence buffer has completed the unloading of the previous sentence to an output device, the presence of the S_{BB} (Status-Buffer Busy) signal instructs the photostore to remain upon the same track upon which a matching entry was found until the sentence buffer signals readiness for acceptance of new information. The 1400 rpm rotational velocity of the photostore glass disc provides an interrogation of the "buffer busy" status at 43 millisecond intervals until buffer acceptance of information may continue. If the output typewriter is in on-line operation, the 10-character per second operational speed of this device will introduce a substantial slowdown of operations. On-line typewriting of the output is therefore valuable only for monitoring, system debugging and equipment demonstration.

Magnetic tape output provides the fastest system operation. The magnetic tape recording time is computed from the following data: 1) 7.5 milliseconds for the tape to be accelerated to a uniform velocity; 2) allowing for spaces between words and punctuation, 140 characters average at 67.5 microseconds per character (operating the 729 II in the low density mode) for a writing time of 9.45 milliseconds; and 3) deceleration and completion of reading while recording as a check upon the correctness of the record written, 3.5 milliseconds. Thus, the time necessary for the complete recording of the sentence unit record would be the sum of the stated component times, viz., 20.45 milliseconds. Using the photostore entry random access figure of 30.0 milliseconds, it may be seen that tape recording of the previous sentence will have been completed before the first word of the next sentence has been located.

To circumvent the system delay imposed by the on-line use of the typewriter when only a system monitor rather than complete hard copy read-out is required, a partial-sentence typewriter operation is available at the discretion of the operator. Preselection keys, numbered one through nine, are used to provide a partial type-out of the stored target language sentence. In this mode of system operation the typewriter would type-out, beginning with the first word of the sentence, the preselected number of words (or the complete sentence if there are less words in the sentence than in the preselection). This is implemented by counting spaces between words until the count is in proper relation to the preselection. To exemplify this mode of operation, the preselection "one" could be made and the typewriter would only type the first word of the sentence; the system would then execute complete sentence storage upon the magnetic tape.

Occasional reference to this type-out of only the first word or first few words of a sentence would serve to monitor proper operation of the translation system.

Reference has been made to the off-line production of hard copy where human readable output is required. This may be accomplished by utilization of the buffer in the magnetic tape-to-typewriter mode, with the buffer system dissociated from the search logic and the photostore. During on-line system operation only magnetic tape would be prepared. Then during off-hours those magnetic tape files for which human-readable copies are required would be printed or punched out by the typewriter. In this case, input to the memory core array is from the magnetic tape rather than from the V_{2D} register of the search logic.

The language translation equipment described in this paper is used for continuing linguistic research and not as a production facility where high throughput is the paramount consideration. The high operating speed of the buffer, when considered in relation to the operating speeds of the output devices, makes possible several different multiplexing modes for the acquisition of higher throughput. For example, during the on-line operation of the system, the output material would be directed to several magnetic tape drives, a complete article or text to a given tape drive. Use of a multiplicity of tape drives represents a minor addition to the system because the buffer was designed to serve ten tape units. The multiplexing of tape drives is presented in terms of storing complete articles upon one tape rather than dispersal of sentences between tapes because, even with the fast random access speed of the photostore, the magnetic tape can record the sentence in less time than the average access time of the first word of the next sentence. Thus, there would be no advantage to time-sharing of the buffer between the output operation of recording upon tape and the input operation of accepting new data from the photostore.

During the off-line production of hard copy, a battery of typewriters would be employed; each typewriter to transcribe the data from a given tape drive. Buffer address assignments would be made with an address block allocation for each tape drive and associated typewriter. Each of the typewriters may then be cyclically supplied with a character to be typed from its respective record. While a given typewriter is typing the character, the buffer is securing the character for the next machine in the battery. Ten typewriters at a time may be served by this extension of the buffer system.

A system projected for completion during 1960 would have all input to the processing system pass through the buffer. This raw input information will then be processed against the photostore and the photostore matching entries read out to the buffer. This read-out, in conjunction with information merged from previous passes, would serve as new data for the next table lookup pass. In this manner iterative processing will be carried out to any depth required by linguistic considerations. The table lookup passes would be classified as to whether the intent of the pass was to secure syntactic or semantic information; the final pass against the photostore would load the output region of the storage preparatory to transfer of data to the output device.

The present translation system permits the modification of a word meaning by the preceding word; this selection of correct meaning is accomplished by the ρ-prefixing conditional addressing technique described in the previous portion of this paper. Storage of the entire sentence in the buffer will permit the construction of elaborate linguistic trees for the resolution of language ambiguities, thus resolving word meanings by succeeding words and by words some distance away. The capacity of the buffer should permit maintenance of a complete "processing trail" of all of the successive passes. This processing trail will be available at any time for off-line linguistic analysis.

The sentence buffer is presently used for the buffering and control of target language output to a magnetic tape unit and an output typewriter. Off-line magnetic tape to typewriter operation is in use for increased efficiency of hard copy production. Integration of the buffer, under the guidance of linguistic research, into more complex sentence analysis routines is now in progress. The combination of the sentence buffer and the photostore-table lookup method is felt to constitute an optimum machine for the processing of natural languages.

VI. SUMMARY

The table-look up language-processing machine described in this article has been constructed and tested, and has already been put to use in translating Russian technical articles and newspapers into rough English. Its present usefulness is the result of a large high-speed random access dictionary and novel addressing features. Most important of its present features are those which

a) determine the longest possible match,

b) condition some searches upon the information retrieved in previous searches.

It is recognized that syntactic and contextual analysis by the machine is a necessary feature for further improvement of the language-processing ability. Development of these analytical techniques is now underway. A lexical buffer memory is already useful for output operations and will allow sentence analysis as soon as the linguistic methods are more completely developed.

This table-lookup machine represents a departure from the usual stored-program type of calculator which has come into wide usage for numerical computation over the past decade. The importance of this type of machine organization is expected to grow as the intellectual community seeks to understand and make use of the preponderance of human thought which is expressed in words, sentences, documents, languages and other forms of ideas. These can be relatively ambiguous when compared with precise symbols and numbers and must, therefore, be analyzed for proper meaning.

BIBLIOGRAPHY

1. King, G. W., Brown, G. W., and Ridenour, L. N. "Photographic Techniques for Information Storage," *Proceedings of the I.R.E.*, Vol. 41, No. 10, October, 1953, 1421-1428.

2. Shiner, G. "The USAF Automatic Language Translator, Mark I," 1958 *IRE National Convention Record*, Part 4, 296-304.

3. Bar-Hillel, Y. "Can Translation be Mechanized?" *Mada*, Vol. 1, No. 2, (April, 1956).

4. Harper, K. E. "The Mechanical Translation of Russian: Preliminary Report," *Modern Language Forum*, 38, (1953), 12-29.

5. Oettinger, A. E. "A Study for the Design of an Automatic Dictionary," Mimeographed, *Harvard* (1953).

6. Dostert, L. *Mechanical Translation of Languages*, Wiley, New York (1955), 124-135.

7. Oswald, V. A. and Fletcher, S. L. "Proposals for Mechanical Resolution of German Syntax Patterns," *Modern Language Forum*, 36, (1958), 1-24.

8. Yngve, V. *Mechanical Translation of Languages*," Wiley, New York, 208-226.

9. Harper, K. E. and Hays, D. G., "The Use of Machines in the Construction of a Grammar and Computer Program for Structural Analysis," *RAND Report* P-1588, January 9, 1959.