NBS TECHNICAL NOTE

413

THE SOLID SYSTEM. II. NUMERIC COMPRESSION
THE SOLID SYSTEM. III. ALPHANUMERIC COMPRESSION

U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards

THE NATIONAL BUREAU OF STANDARDS

The National Bureau of Standards[1] provides measurement and technical information services essential to the efficiency and effectiveness of the work of the Nation's scientists and engineers. The Bureau serves also as a focal point in the Federal Government for assuring maximum application of the physical and engineering sciences to the advancement of technology in industry and commerce. To accomplish this mission, the Bureau is organized into three institutes covering broad program areas of research and services:

THE INSTITUTE FOR BASIC STANDARDS . . . provides the central basis within the United States for a complete and consistent system of physical measurements, coordinates that system with the measurement systems of other nations, and furnishes essential services leading to accurate and uniform physical measurements throughout the Nation's scientific community, industry, and commerce. This Institute comprises a series of divisions, each serving a classical subject matter area:

—Applied Mathematics—Electricity—Metrology—Mechanics—Heat—Atomic Physics—Physical Chemistry—Radiation Physics—Laboratory Astrophysics[2]—Radio Standards Laboratory,[2] which includes Radio Standards Physics and Radio Standards Engineering—Office of Standard Reference Data.

THE INSTITUTE FOR MATERIALS RESEARCH . . . conducts materials research and provides associated materials services including mainly reference materials and data on the properties of materials. Beyond its direct interest to the Nation's scientists and engineers, this Institute yields services which are essential to the advancement of technology in industry and commerce. This Institute is organized primarily by technical fields:

—Analytical Chemistry—Metallurgy—Reactor Radiations—Polymers—Inorganic Materials—Cryogenics[2]—Office of Standard Reference Materials.

THE INSTITUTE FOR APPLIED TECHNOLOGY . . . provides technical services to promote the use of available technology and to facilitate technological innovation in industry and government. The principal elements of this Institute are:

—Building Research—Electronic Instrumentation—Technical Analysis—Center for Computer Sciences and Technology—Textile and Apparel Technology Center—Office of Weights and Measures —Office of Engineering Standards Services—Office of Invention and Innovation—Office of Vehicle Systems Research—Clearinghouse for Federal Scientific and Technical Information[3]—Materials Evaluation Laboratory—NBS/GSA Testing Laboratory.

---

[1] Headquarters and Laboratories at Gaithersburg, Maryland, unless otherwise noted; mailing address Washington, D. C., 20234.

[2] Located at Boulder, Colorado, 80302.

[3] Located at 5285 Port Royal Road, Springfield, Virginia 22151.

UNITED STATES DEPARTMENT OF COMMERCE
Alexander B. Trowbridge, Secretary
NATIONAL BUREAU OF STANDARDS • A. V. Astin, Director

TECHNICAL NOTE 413

ISSUED AUGUST 15, 1967

THE SOLID SYSTEM. II. NUMERIC COMPRESSION

P. A. D. deMaine,* K. Kloss,** and B. A. Marron*

THE SOLID SYSTEM. III. ALPHANUMERIC COMPRESSION

P. A. D. deMaine,* B. A. Marron,* and K. Kloss**

* Center for Computer Sciences and Technology
Institute for Applied Technology
National Bureau of Standards
Washington, D.C. 20234

** Applied Mathematics Division
Institute for Basic Standards
National Bureau of Standards
Washington, D.C. 20234

For sale by the Superintendent of Documents, U.S. Government Printing Office, Washington, D.C., 20402
Price 30 cents.

FOREWORD

THE SOLID SYSTEM. II. NUMERIC COMPRESSION
THE SOLID SYSTEM. III. ALPHANUMERIC COMPRESSION

Two independent computer algorithms designed to reduce greatly the storage requirements for information are described in this two-part note.

The Numeric Compressor is a fully automatic scheme for achieving substantial savings in both fast and slow storage by storing numerical data in a highly compact form after its conversion to fixed point format. The algorithm uses truncation, differencing and packing in a unique fully automatic scheme with provision for automatic decompression. The amount of compression achieved is determined by the "lowest limit of significance," the range and the sequential pattern of the data to be stored.

The Alphanumeric Compressor is a recursive bit-pattern recognition technique for compressing any kind of information in a precisely reversible way. It is language and content independent, and can operate on information which has already been compressed by the Numeric Compressor. It automatically stores with the compressed information all the data needed for the system to decompress the information on demand back to its original form. Limited tests have demonstrated substantial savings.

Both compressors were implemented initially on the National Bureau of Standards Pilot Data Processor, and are currently being reprogrammed for a second computing system.

CONTENTS

LIST OF FIGURES

CONTENTS (Continued)

Part I

THE SOLID SYSTEM. II. NUMERIC COMPRESSION

P. A. D. deMaine
K. Kloss
B. A. Marron

This part of NBS Technical Note 413 describes the general NUMERIC COMPRESSOR (NUPAK) Algorithm for automatically compressing (encoding) or decoding compressed numerical information, which may of course have come from graphical information. The amount of compression achieved is determined by the "lowest limit of significance," the range, and the sequential patterns of the data to be stored. The encoded information can be stored in memory or on external storage devices in a small fraction of the space normally required, and can be expanded (decoded) item-by-item whenever needed by the system.

Key Words:
numeric compression, information handling, high-speed information transmission, information storage and retrieval, systems analysis.

1. INTRODUCTION

The ever increasing amount of information generated in the scientific disciplines and the current attempts to devise and implement truly large-scale computer-based retrieval systems together demand that methods be found for increasing the "informational content" per unit of external (slow) and internal (fast) memory. The ultimate physical limitations on the speed and storage capacity attainable with computers are constant reminders that, no matter how large or fast a computer configuration is, its full potential for processing very large amounts of information (e.g., in a National Information System) can only be achieved if ways are found to increase substantially the "informational content" of whatever storage is available.

In this regard it should be noted that even with the largest configuration now in existence there is scarcely sufficient storage for coding more than a few thousand 500-page volumes [1]. Even with the projected "large" memory units [2.a.], the amount of coded information that can be stored is increased by not more than two or three orders of magnitude. While unlike some authors [2.b.], we do not

presuppose that the "universality of information" is to be stored on a single configuration, it is desirable that the cataloging and file structure components should at least fit onto a single configuration. Similarly, while implementation of the ZVA/FVA Compressor Algorithms [3], which contain a small complete hybrid data compressor for reducing redundant signals, will increase the "informational content" of transmitted data, it leaves largely unaltered the problem of further compressing the output received from a space telemetry system.

In very large-scale computer-based retrieval systems, it is essential that each document or item of information be stored in the smallest space possible. In existing retrieval or storage systems this is achieved by omitting bulky items of information or, in some cases, restricting coding to selected features like band positions and associated intensities (for spectra). It is the experience of one author (deMaine) that the resultant loss of detail (e.g., in spectroscopy, of band shapes, shoulders and other details which are omitted) at best severely restricts the usefulness of these systems as research tools.

In the ultimate analysis the informational content of fast memory and the rate at which required data can be transferred from slow to fast memory are two of the several definitive factors affecting the efficiency of any large retrieval system. Quite obviously, the bookkeeping tasks and programming associated with file organization and search procedures can be simplified if the "rate of transmission of required information" (RTRI) through the computer is increased. This RTRI and thus the efficiency of the system will be increased if the information is in compressed form. An automatic decoder, capable of item-by-item expansion of the compressed information as it is required, would be an essential part of such a system. In this report algorithms for compressing (encoding) and for decoding compressed numerical information are discussed. Alphanumeric compressors and their combination with the numeric compressor are described in Part II of this Technical Note. Here it is supposed that a large number (i.e., > 50) of floating or fixed point numbers are to be stored.

With the BINARY COMPRESSOR-EXPANSION routines now generally available [4], which store twenty or more machine words per 80 column card (or the equivalent on magnetic tape), external (slow) storage requirements are reduced by as much as 95%. However, these routines do not by themselves reduce the internal (fast) storage requirements. With large files, this inevitably means that there is a complex structure with a large number of "instruction words" prefacing the file and its component sub-files, and the complexity of the search procedure is increased. The BIN PROCEDURE [5], for storing compressed numerical information on punched cards on magnetic tape, can be used in conjunction with the Self-Judgment Method of Curve-Fitting [6] or its automatic version [5], to effect substantial savings in internal (fast) storage, and, in conjunction with the binary routines, can achieve additional savings in external storage. However, these methods [5,6], are not

suitable as a basis for a general automatic data compressor because of their complexity and, in the curve-fitting methods, the requirement that the data fit a simple continuous function.

Here there is described the NUMERIC COMPRESSOR(NUPAK) Algorithm for automatically compressing (encoding) or expanding (decoding) numerical information on a machine with binary arithmetic capability. With NUPAK alone, internal or external storage requirements can be reduced, under the most favorable circumstances, by as much as 32,700:1. The actual amount of compression achieved depends on the kind of data to be stored and the number of significant figures to be retained. In the latest version of NUPAK, savings of more than 90% appear to be average. When NUPAK is used in conjunction with a BINARY COMPRESSION routine [4] and the ALPHANUMERIC COMPRESSOR (see below), storage requirements are reduced by another substantial factor. By altering the coding base (i.e., byte length) of the information after compression[1] another substantial saving in the internal storage may be achieved.

The new capability permits storage in the system of numerical data normally excluded because of their volume. Thus, for example, it is now feasible to write programs which would permit direct computer searches of spectra and other physical data which are frequently used to "fingerprint" chemical compounds and biological or bacteriological processes. Moreover, the storage required for even the most detailed "fingerprint" is frequently less than the amount required to store the distinctly incomplete and inadequate partial information typically offered.

The NUMERIC COMPRESSOR (NUPAK) described here can be used either separately as a compressor of numeric information or in conjunction with the ALPHANUMERIC COMPRESSOR [7] and the BINARY COMPRESSOR (IOPAK). In the proposed Self Organizing Large Information Dissemination (or Retrieval) System (SOLID SYSTEM) [8], NUPAK, ANPAK, and IOPAK are conceived as the components of the COMBINED COMPRESSOR (COPAK) [7] for compressing data and programs.

---

[1] The particular algorithms and computer programs described in this Technical Note have been largely superseded by progress made since this manuscript was prepared, although the major concepts remain the same.

2. DESCRIPTION OF PRINCIPLES

Basic to the NUMERIC COMPRESSOR (NUPAK) is the concept that all numerical data in the experimental natural sciences have some limit to their significance, imposed either by the conditions of their collection or by the context in which they are to be used. For any observable or variable with ρ values there is always one lowest "Limit of Significance"(LS) which defines the maximum reliability of all ρ items adequately. Moreover, in those cases where exact (i.e., LS = 0) and experimental or mathematical but inexact (LS $\neq$ 0) information are interrelated by some unknown function, it is always possible to obtain a nonzero LS value either by projecting the LS for nonexact data onto the exact scale[1] or by using the number of significant figures in the "exact" data to determine its accuracy.

This information (LS) is used in the compressor to rescale the floating point numbers in fixed point. In most cases the LS value is set by the known limit of reliability of the collecting device or technique. For example, there is no infrared absorption spectrophotometer generally available that can measure unexpanded transmittances and frequencies of solutions to better than one percent and one reciprocal centimeter respectively. Thus LS values of 0.5 percent for absorbance and 0.5 cms^{-1} for frequency scales will adequately define the significance of infrared spectral data collected with any

---

[1]This procedure is described for SERM(J) in reference 5.

conventional instrument. The data regenerated from the compressed form will always be accurate to well within the lowest limit of significance.

The compressive calculations involved in NUPAK can be summarized as truncation (to fixed point), scaling, repeated differencing, and packing. These calculations will be outlined here and described fully in the next Section. Suppose that there are values (x_i) of some observable x with a lowest Limit of Significance (LSx) and a minimum value XMIN. The following definitions are peculiar to the NUMERIC COMPRESSOR (see Figure 1):

BIN WIDTH (BWx) equals LSx/2.0 > 0.0

BIN VALUE (BVx) associated with x_i is $BVx_i = (x_i - XMIN)/BWx$. BVx_i is truncated without rounding.

TRUNCATED BIN VALUE (TBVx) associated with x_i is computed from the integer BVx_i value thus:

$$\left.\begin{array}{ll} \text{For } i = 1 & X_1 = BVx_1 \\ \text{For } 1 < i \leq \rho & X_i = BVx_i - BVx_{i-1} \end{array}\right\} \ldots A$$

DEPTH OF REPRESENTATION (NDR) The magnitude of the values to be compressed can sometimes be decreased by successive applications of equations (A), with BVx_i values replaced by X_i. The Depth of Representation (NDR = $\overline{NDR}$ + 1) is computed from the number ($\overline{NDR}$) of recursive applications of equations (A). In the programmed procedure the computer in effect selects that value for NDR which will give the most compression when the fixed point numbers (X_i) are packed. This differencing procedure is somewhat similar to techniques used earlier [9] for noncomputer reductions of tables.

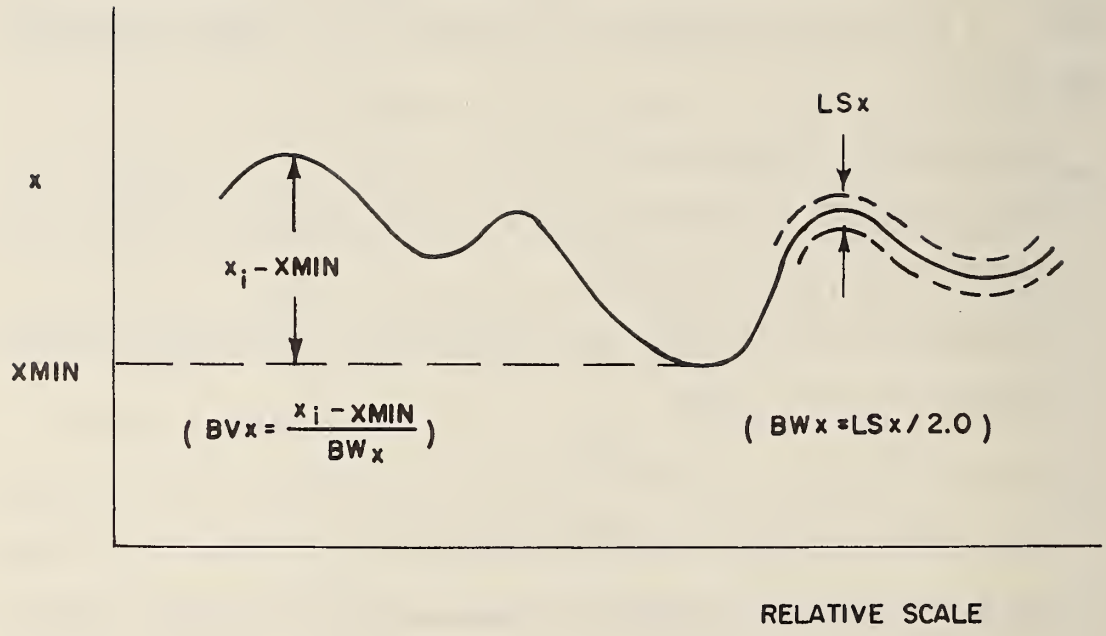


Figure 1. Illustration of the definitions in the NUMERIC COMPRESSOR (NUPAK). Bin values (BVx) are truncated and stored in compressed form.

To regenerate the original information from the final $TBVx_i$ values (obtained after (NDR - 1) recursive applications of equations (A)), together with XMIN and BWx, the recursive procedure is reversed to obtain new BVx_i values (say $\overline{BVx_i}$). New x_i values ($\bar{x}_i$) are computed thus:

$$\bar{x}_i = (\overline{BVx_i} + 0.5)\ BWx + XMIN.$$

Since $|\bar{x}_i - x_i| \leq BWx/2.0 = LSx/4.0$, the regenerated values ($\bar{x}_i$) are well within the lowest Limit of Significance (LSx) of the original values (x_i). The term 0.5 in the above equation halves the maximum error which can occur in truncating the bin values.

Even with the common small variable-word-length machines and higher coding languages (e.g., FORTRAN), the saving in internal and external storage is substantial if $\rho > 4$. However, the truly significant savings are achieved when the new information (XMIN, BWx, $TBVx_i$,.....) is packed in a binary arithmetic machine. The packing (or encoding) procedure is described in the next Section.

3. NUMERIC COMPRESSOR ALGORITHM (NUPAK)

NUPAK was initially implemented on the Pilot Data Processor at the National Bureau of Standards, which has 32,736 sixty-eight (Boolean) or sixty-five (arithmetic) bit words of memory storage and three addresses per instruction word. Among several unique features is a TRANSPLANT SEGMENT WITH SHIFT (TL) instruction which is especially useful in assembling composite words. Modifying parameters for the TL instruction define the magnitude of the shift and the segment which is to be transplanted. Here it is supposed that the equivalent of the acyclic TL instruction and arithmetic words with N binary bits are available on the computer to be used, and that no $|TBVx_i|$ value to be

stored requires more than (N/2 - 4) bits. The truncated bin values ($TBVx_i$) with depth of representation NDR are to be compressed into the array STCF.

A. STRUCTURE OF COMPRESSED INFORMATION (ARRAY STCF)

The left-most six bits of each word of array STCF contain three items (S, F and NS) which define the amount of information and type of compression in the word and whether or not the next sequential word contains further $TBVx_i$ information. In the encoding procedure the computer automatically deduces those S, F and NS values which give the greatest compression. In the decoding procedure this information (S, F and NS) is used by the computer to expand the compressed information to its original form. (See Table I.)

S (Sign Bit N)

0 The next sequential word of array STCF also contains $TBVx_i$ information.

1 The last $TBVx_i$ value is contained in this word.

F (Bit N-1)

0 There is a number (NS, with NS $\geq$ 3) of equal segments, each with the integral part of [(N-6)/NS] bits. The first three segments specify (n-1) equal $TBVx_i$ values (one is $TBVx_{I+1}$) after $TBVx_I$. The fourth to fifteenth segments contain the designated values of $TBVx_i$.[1]

[1] In the latest version the identification and packing of sequences of identical numbers are more efficient. The incompleteness of compression apparent in the method described here is remedied.

TABLE I

Illustration of encoding procedure by the NUPAK algorithm. Input consists of SOS (=0), JI (=18), LSx (=0.020) and eighteen x_i values. XMIN (=1.010) and BWx (=0.010), computed in Step 1, are used to calculate the BVx_i; then these are used to compute $TBVx_i$, and NDR. The final information (SOS, NDR, XMIN, $TBVx_1$, and BWx) is stored in six 65 bit computer words (Step IV). NDR, SOS, S, F, and NS are codes automatically deduced (and in decoding, executed) by the computer.

x_i	BVx_i	$TBVx_i$		x_i	BVx_i	$TBVx_i$	
		NDR=2	NDR=3			NDR=2	NDR=3
1.010	0.0	0	0	35.010	3400.0	- 200	- 300
2.015	100.5	100	100	40.011	3900.1	500	700
4.011	300.1	200	100	43.012	4200.2	300	- 200
7.013	600.3	300	100	50.011	4900.1	700	400
11.014	1000.4	400	100	70.010	6900.0	2000	1300
16.011	1500.1	500	100	71.015	7000.5	100	-1900
22.010	2100.0	600	100	66.011	6500.1	- 500	- 600
29.011	2800.1	700	100	80.012	7900.2	1400	1900
37.012	3600.2	800	100	80.013	7900.3	0	-1400

Step IV (Encoded Output)

SOS = -NDR = -3

BWx = 0.010

XMIN = 1.010

TBVx values are stored in 3 words from high order to low order address. (Decimal $TBVx_i$ are shown)

S	F	NS	Segment 1	Segment 2	Segment 3		
1	1	3	-600	1900	-1400	0	0
65,64,63-60,59				41,40	22,21		3,2,1

S	F	NS	Segment 1	Segment 2	Segment 3	Segment 4			
0	1	4	-200	400	1300	-1900	0	0	0
65,64,63-60,59				46,45	32,31	18,17			4,3,2,1

S	F	NS	Segment 1	Segment 2	Segment 3	Segment 4	Segment 5				
0	0	5	0	100	8	-300	700	0	0	0	0
65,64,63-60,59				49,48	38,37	27,26	16,15				5,4,3,2,1

Bit structure of Segment 4 in the last word.

-300

1	0	1	0	0	1	0	1	1	0	0
11	10	9	8	7	6	5	4	3	2	1

Suppose with N = 65, optimum compression was achieved by storing the information for the (n+4) values $TBVx_I$, (n-1) $TBVx_{I+1}$ values, $TBVx_{I+2}$, $TBVx_{I+3}$, $TBVx_{I+4}$ and $TBVx_{I+5}$, thus:

Bit N-5	N-6							1
S,F=0, NS=7	$TBVx_I$	$TBVx_{I+1}$	(n-1)	$TBVx_{I+2}$	$TBVx_{I+3}$	$TBVx_{I+4}$	$TBVx_{I+5}$	Excess Bits

Here there are seven equal segments. F = 0 indicates that the first three segments contain all the information for the first n values.

1 The NS values of $TBVx_i$ are stored sequentially in the NS equal segments, each with the integral part of [(N-6)/NS] bits. The word is then shifted left the number of excess bits, which equals(N-NS times the number of bits/segment).

NS (The four bits N-2 to N-5)

The number of segments (NS) is automatically computed from the available space (N-6 bits) and the number of $TBVx_i$ values that can actually be stored safely in the word of array STCF. Involved in this procedure is a bit count of twice the largest absolute[1] item (i.e., for the example above, these items would be (n-1), $TBVx_I$, $TBVx_{I+1}$, $TBVx_{I+2}$, ...) and a consideration of the alternate

---

[1] To allow for the signs of the items to be stored.

(F = 0 or 1) forms of compression. The left-most bit in each segment designates the sign of the item.

Additional compression can be achieved by redefining the commands F and NS so that they appear in only the first word of a compressed string. In the algorithm described here the commands defined above appear in each word.

B. STATUS OF SYSTEM COMMANDS (SOS)

A single word with each string (SOS) notifies the computer of the status of the string in its memory. Initially SOS is entered with the data to be stored as a command. After executing the command, SOS is redefined and, together with the compressed or expanded string, is stored in the file. In the COMBINED COMPRESSOR (COPAK) [7] SOS is again redefined and the string is further compressed by a recursive bit pattern recognition technique; then binary compression occurs before the information is stored in an external device. In retrieval operations the computer uses the redefined SOS to determine what steps must be taken if the information is required in its original form.

Input Commands (SOS)

0 The numerical information (floating or fixed point) is to be compressed, if possible, by NUPAK. SOS is changed to -NDR (compression achieved) or -Ó (no compression).

1 This information is not to be compressed in the computer. The Binary Compressor will be used before transferring it to the file (magnetic tape or cards).

SOS is unchanged.

-1 The information is to be compressed with the ALPHANUMERIC component [8] before storage. SOS is changed to -0.

Automatic Command (SOS)

The commands given here are those associated with the NUMERIC COMPRESSOR (NUPAK). (Those associated with theALPHANUMERIC component are described in Part II of this Technical Note.)

-NDR Decode (expand) this compressed information if it is required. The DEPTH OF REPRESENTATION is NDR.

1 or -0 This information has not been compressed with NUPAK.

C. ILLUSTRATION OF COMPRESSION

To illustrate the fully automatic encoding procedure, consider the information in Table I. The following step-wise procedure occurs:

Step I: From the input (x_1, LSx and SOS) the computer calculates BWx and XMIN, if SOS = 0.

Step II: The BIN VALUES (BVx_1) associated with each x_1 are computed.

Step III: The recursive application of equations (A) yields the DEPTH OF REPRESENTATION (NDR) and the final values for the TRUNCATED BIN VALUES ($TBVx_1$). SOS is set equal to -NDR. The actual procedure for calculating NDR is described in the next Section.

Step IV: The $TBVx_1$ values are stored in compressed form in array STCF. This information, together with SOS, XMIN and

BWX, is stored in the external device.

The compressed form of the 19 machine words normally required to store the 18 x_1 values in Table I is six machine words. (See Table I). For much larger samples, the compression can be as high as 32,700 to 1 (for sequential numbers.) A typical compression is shown in Table II on page 23.

D. FLOW CHART

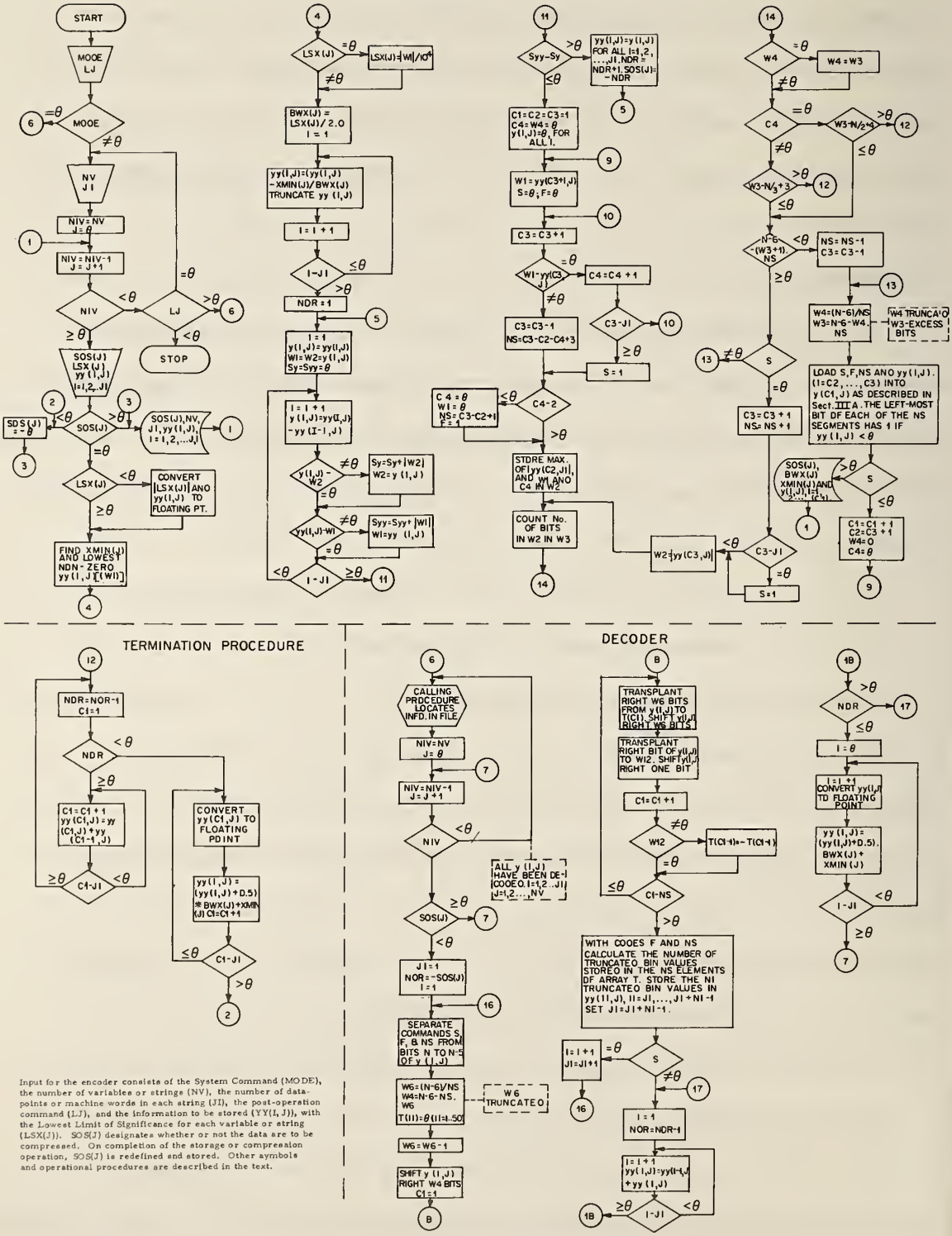
In the operational form of the NUPAK algorithm (Figure 2) the respective arrays YY and Y are used to store the decoded (expanded) and encoded (compressed) numerical information for each variable. YY is also used to store intermediate results. In the encoder part of the NUMERIC COMPRESSOR NV, JI and LJ are defined thus:

NV is the number of single sets of data to be processed.

JI is the number of items in each set of data.

LJ is the post-operation command.

NIV and J, the two variable counters, are set equal to NV and 0, respectively. Next NIV and J are respectively decremented and incremented by one, and if NIV <0 control passes to the post-operation counter (LJ). With LJ control is passed to the READ FILE (LJ = 0) or to the SEARCH PROCEDURE (LJ > 0) or STOP (LJ < 0). If NIV ≥ 0 the data for the Jth variable or string are loaded in arrays YY, SOS and LSX. If SOS <0 it is changed to -0, and after the expanded information is stored in the file control again passes to the procedure for finding the next string; otherwise, either the information is stored in the file in uncompressed form (SOS(J) >0) or the compression begins (SOS(J)=0).



Input for the encoder consists of the System Command (MODE), the number of variables or strings (NV), the number of data-points or machine words in each string (JI), the post-operation command (LJ), and the information to be stored (YY(I,J)), with the Lowest Limit of Significance for each variable or string (LSX(J)). SOS(J) designates whether or not the data are to be compressed. On completion of the storage or compression operation, SOS(J) is redefined and stored. Other symbols and operational procedures are described in the text.

FIGURE 2. Flowchart for the fully automatic NUMERIC COMPRESSOR (NUPAK)

If compression is desired (SOS(J) = 0), the minimum value of YY(I,J) (I = 1, 2, ..., JI) is computed and stored in XMIN(J). If the lowest Limit of Significance (LSX(J)) is less than zero.(i.e., it was entered 0 because the data were in fixed point), the XMIN(J), LSX(J) and the YY(I,J) values are converted to floating point. If LSX(J) is equal to 0.0, it is set equal to the lowest nonzero value of YY(I,J) divided by 10000.0, and XMIN(J), which may be zero, is computed. The BIN WIDTH (BWX(J)) is computed from $|LSX(J)|$; then the BIN VALUES (see definitions) are computed, truncated and stored in array YY(I,J). The following procedure is executed to determine the DEPTH OF REPRESENTATION (NDR).

Step I: NDR is incremented by one from an initial value of zero.

Step II: TRUNCATED BIN VALUES are computed from the YY(I,J) values with equations (A) and stored in Y(I,J).

Step III: Absolute values of YY(I,J) and Y(I,J) are summed (over I = 1, 2,..., JI) and stored in SYY and SY respectively. The summing procedure is as follows: If two or more consecutive values in either array are equal, then only the first one is included in the sum. If SY <SYY, information in array Y is transferred to array YY and Steps I and II are repeated. If SYY ≤ SY, SOS(J) is set equal to -NDR and control passes to the compression procedure. (This summation and comparison procedure closely approximates a comparison of storage requirements for two fixed point arrays.)

In the compression procedure, four counters (C1, C2,C3 and C4) and four temporary storage locations (W1, W2, W3 and W4) are used to store the TRUNCATED BIN VALUES of YY(I,J) in Y(I,J) in the most compressed form possible. If any economy in storage cannot be effected, a situation designated in Step V below by TERMINATE, SOS(J) is set equal to minus zero and the floating point values of YY(I,J), within the "lowest limit of reliability" (Section 2), are recomputed from the DEPTH OF REPRESENTATION(NDR), YY(I,J), BWX(J) and XMIN(J) as described earlier. These values are stored in the file in the uncompressed (expanded) form.

The compression procedure is described next. The number of bits per machine word is donoted by N. Four counters (C1,C2, C3 and C4) and four words W1, W2, W3 and W4) are used as follows in the compression procedure:

Y(C1,J) is the machine word into which the information computed with C2, C3 and C4 is to be packed (or compressed).

YY(C2,J) is the number that will be packed into the left most segment of Y(C1,J).

C3-- is an index used to scan elements of array YY(I,J).

(I > C2) for consecutive equal values.

C4-- counts the consecutive equal terms after the first term of array YY(I,J) (i.e., I > C2) scanned with index C3.

W1-- is the term YY (C2 + 1, J) used to scan the array YY(I,J) for consecutive equal terms. W2, W3 and W4 are defined in Steps IV and V.

Step I: Set counters C1 and C2 to one, C4 to zero and W4 plus the JI elements of array Y(I,J) (here J is fixed) to

zero.

Step II: Set C3 = C2 + 1. If C3 > JI, decrement C3 and set W1, S and F to zero, and go to Step IV. If C3 ≤ JI, set W1= YY(C3,J) and S and F to zero. For C3 < JI go to Step III; otherwise go to Step IV.

Step III: Increment C3 and compare with JI. If C3 > JI, go to Step IV; otherwise compare W1 and YY(C3,J) . If YY(C3,J) =W1, increment C4 and repeat this step. If W1 ≠ YY(C3,J), decrement C3 and go to Step IV. W1 contains the common entry of array YY(I,J) with C2 < I ≤ C3.

Step IV: If C4 ≤ 2, set W1 and C4 to zero and F = 1. Next find the largest absolute value among W1, YY(C2,J) and C4, and store it in W2. A bit count of the absolute value of W2 in Step V discloses the number of bits so far known to be required for each segment of Y(C1,J).

Step V: In W3, count the number of bits (exclusive of the sign) required to store W2. If W4 = 0, set W4 = W3. If W3 < W4, set W3 = W4. TERMINATE if W3 > N/3 - 3 (for C4 ≠ 0) or W3 > N/2 - 4 (for C4 = 0); otherwide to to Step VI. W4 + 1 is the last known number of bits per segment needed to safely pack information as currently known into Y(C1,J). If (W3 + 1) · NS exceeds the number of bits available in Y(C1, J) for packing (see Step VII), then in Step IX, NS is decremented

and W4 is used in place of W3.

Step VI: Compute the number of segments (NS) required to pack the information as far as currently known into Y(C1,J). If C4 = 0, NS = (C3 - C2 + 1). For C4 ≠ 0, NS = (C3 - C4 - C2 + 3). Go to Step VII.

Step VII: If NS·(W3 + 1) exceeds (N - 6), go to Step IX; otherwise go to Step VIII.

Step VIII: Increment C3 and NS. If C3 > JI, go to Step IX: otherwise set W2 = |YY(C2,J)| and go to Step V.

Step IX: Decrement C3 and NS by one. With NS, compute the number of bits (W4 = (N - 6)/NS) per segment of the compressed word and the number of excess bits (W3) it contains. Use values of C1, C2, C3, C4, W3 and W4 to compress the information in YY(I,J) with C2 ≤ J ≤ C3 into Y(C1,J), as described earlier. (See definition of F on page 8. The left-most bit of each segment is used to indicate the sign of the stored item (i.e., YY(I,J), W1 and C4). If C3 <JI, set C2 = C3 + 1, C4 = 0, W4 = 0, and C1 = C1 + 1; then go to Step II. If C3 ≥JI, the compression of the original JI numbers into array Y(I,J), with 1 ≤ I ≤ C1, has been completed and Y(C1,J) is set negative.

In the procedure just described, the sign-bit (S) in word Y(C1,J) is set equal to one if C3 = JI, and the compressed information (Y(I,J), I =1, 2, ..., C1), SOS(J), XMIN(J) (the minimum value of the original YY(I,J)) and BWX(J) is stored in the file. This procedure is repeated

until the variable counter (NIV) is less than zero, when control goes to counter LJ. In the file itself, the number of variables or strings (NV) to be stored prefaces the information.

If the counter LJ is greater than zero, control passes to the calling procedure which then determines the location of the desired information. This component is not operable in the current version of NUPAK. Once the file information has been located, the decoder part of the NUMERIC COMPRESSOR is automatically executed as follows. The two variable counters (NIV and J) are set equal to NV and 0 respectively. Next Steps I and XI are executed.

Step I: Decrement NIV and increment J. If NIV < 0, all information has been decoded and control passes to the calling procedure for the next instruction. To achieve decoding of selective items, NV, which is stored with the compressed and uncompressed data, J and NIV would be changed with instructions in the calling procedure. If NIV ≥ 0, control goes to Step II.

Step II: If SOS(J) = -0 or +1, no compression was made. The uncompressed information from the file is stored in whatever temporary storage area is selected (not shown) and control returns to Step I. If SOS(J) < 0, NDR (= -SOS(J)) is not zero, and control goes to Step III.

Step III: The depth of representation (NDR=-SOS(J)) is computed and the decompression of Y(I,J) begins. This is accomplished with the aid of three counters (JI,I

and C1); the words W4, W6 and W12; and a column array (T) with 50 elements. Details of the decoding procedure are shown in the FLOW CHART (See Figure 2). They can be summarized as unpacking each element in array Y(I,J) to array YY(I,J) via the 50 buffer words (array T). When all compressed words for the substring (located in array Y(I,J)) have been unpacked (into YY(I,J)), the differencing procedure (equations (A)) is reversed and then the JI fixed point numbers, located in array YY(I,J), are converted to floating-point by the method described earlier.(See Section 2).

Step IV: The commands S, F and NS (stored in bits N to N-5) are separated from Y(I,J) with the aid of the TRANSPLANT SEGMENT WITH SHIFT (TL) instruction. The number of bits in each of the NS segments (W6) is computed and the excess bits (W4), which contain no information, are eliminated by a right-shift of W4 bits. Initialize C1 = 1.

Step V: With the TL instruction the right-most (W6-1) bits of Y(I,J) are transplanted to T(C1) and bit W6 is transplanted to W12. Y(I,J) is shifted right W6 bits.

Step VI: The sign of T(C1) is set by the contents of W12 (i.e., 0 indicates positive). C1 is incremented and compared with NS. If C1 $\leq$ NS, Step V is repeated; otherwise, Step VII is executed.

Step VII: With the aid of the commands F and NS and the counter JI, the information in array T is transferred to array YY. At the end of this operation, JI = number of expanded items of information.

Step VIII: The sign-bit (S) of Y(I,J) is inspected. If Y(I,J)<0, (i.e., S = 1), all compressed information for the Jth variable has been processed (go to Step IX); otherwise, I and JI are incremented and Step IV is again executed.

Step IX NDR is decremented by one. If NDR = 0, control goes to Step X; otherwise, the recursive procedure in equations (A) are reversed NDR times to recompute the original integer BIN VALUES (BVx_i) in equations (A). These BVx_i values are stored in array YY.

Step X: With the BIN WIDTH (BWX(J)) and MINIMUM (XMIN(J)), the original information (accurate to within BWX(J)/2.0) is computed from the BIN VALUES (stored in YY(I,J)) and stored in array YY.

Step XI: Control passes to Step I.

4. USE OF NUPAK ALGORITHM

To illustrate an actual savings in storage that can be achieved with NUPAK, 561 data-points from the standard polystyrene film infrared spectrum (Figure 3) were compressed to the 68 words in Table II. Note that the 561 transmittance values were compressed to 63 words; the 561 regularly spaced frequency values were compressed to only five words. Considerably less than one second of Pilot machine time was required. Data-points were selected at every 10 cms^{-1} between 4,000 and 2,000

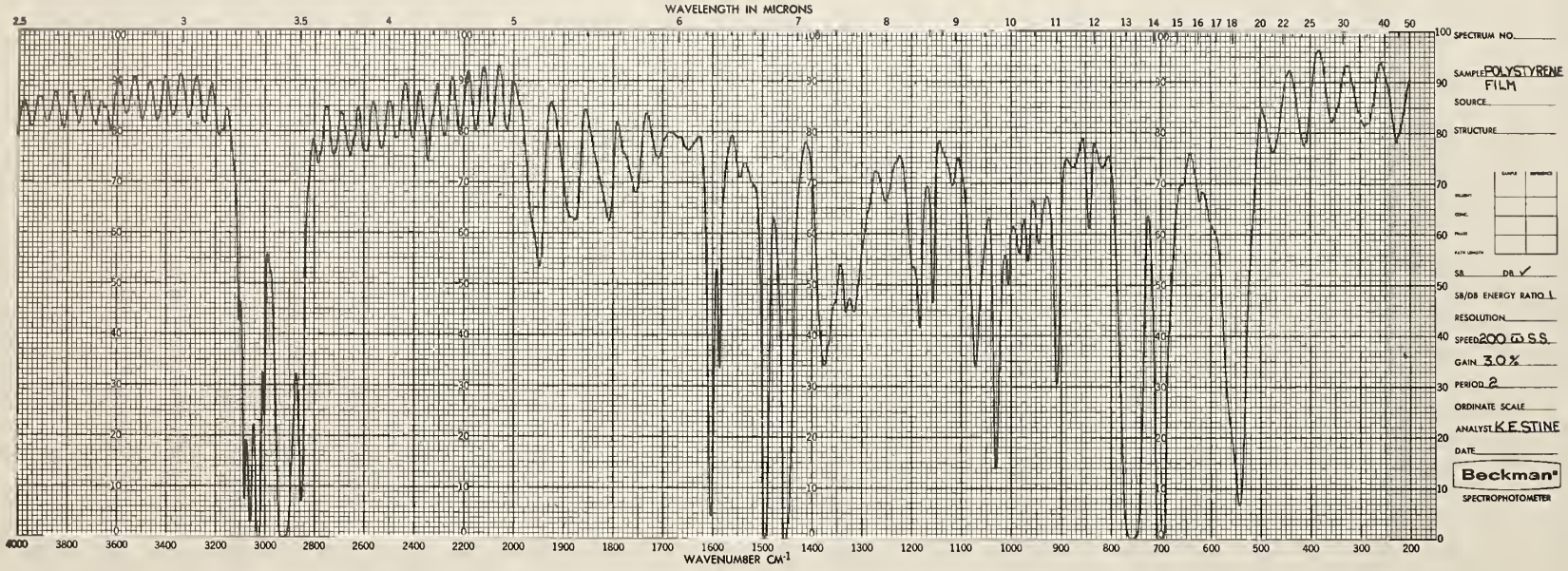


Figure 3. Standard polystyrene film spectrum, taken with the Beckman IR-12 infrared spectrophotometer. (Courtesy of Beckman Instruments, Inc.)

TABLE II

Compressed information for 561 data points taken from the standard spectrum in Figure 3. Symbols have been defined in the text. SOS = -(Depth of Representation). The Lowest Limits of Significance (LSX) are one cm^{-1} and one percent of the transmittance scale. Hexadecimal coding is used in the 65-bit PILOT arithmetic word. Less than one second of PILOT time was required for compression.

THE 561 TRANSMITTANCE VALUES ARE COMPRESSED TO:

```
01 0000 0000 0000 0002    SOS
08 8000 0000 0000 007F    BWX=0.5
00 0000 0000 0000 0000    XMIN
```

SIXTY WORDS OF COMPRESSED INFORMATION

```
01 A003 00A0 1C02 0030
00 D0D2 524A 74B6 3220
00 E933 2056 EED5 C898
00 C120 81C1 944D 1080
00 C492 5144 A0B1 4140
00 D5AB 5908 8842 0210
00 BA03 0586 121C 2810
00 C65D 5145 155A 0AC0
00 CD2A 39CE 9495 3800
00 C205 0608 8305 24C0
00 B000 0002 0060 9080
00 B0C0 F1B0 6C8C D6A0
00 BE8B D461 4102 0420
00 C4B1 8210 5257 87C0
00 C042 0A0E 1996 B440
00 BEB1 6A90 0483 0A08
00 C434 5031 02C5 4E80
00 C2A1 4730 844D 52E0
00 B604 E8E9 3094 1450
00 C65B 4C62 8518 2CE0
00 C473 8240 31CD 5600
00 BC1D A356 0787 0A18
00 C596 591C 5A2E C120
00 D9D6 DD1A 311D 6800
00 D167 2221 6540 CE50
00 D72C 52A8 8640 B760
00 C186 061C 74DD C680
00 B6E0 0020 6151 50D0
00 B410 D1B1 7414 54B0
00 B414 1C1C 0C3C A770
00 C2C4 0C14 93CA 1860
00 B665 98B9 A0B5 39A8
00 C470 8008 4083 3600
00 EE77 50A1 9084 C848
00 D5A3 3A56 6233 8E20
00 C492 4940 4820 34C0
00 C0A7 2C1C 734B 2CC0
00 D125 2CDF 1AB4 A200
00 B9E9 1BA4 8320 58A0
00 CDC4 944A B3C6 B800
00 D294 6B72 C471 A9E0
00 D2A7 BC8D 0A26 3940
00 C4C8 0C4C B54C 1180
00 D214 1DEC 6845 31A0
00 D0AD 9A0D 479E A630
00 D5AD 23B2 38C0 0CC0
00 D60C A363 1422 8940
00 BE49 CC3F 1214 0E08
00 BDE0 0001 0A28 5060
00 B20C 83FC 344D 0510
00 B5B3 ED08 9865 2000
00 C024 C345 5575 C680
00 D129 8B09 059E AD10
00 D5B1 2328 35C5 4A80
00 D583 6303 1812 0C20
00 D28D 19DA 6538 E980
00 D0A9 6A10 D194 7040
00 D5AA 3646 B498 4C40
00 D010 405A 7121 4740
00 AA94 0A02 8141 2080
```

THE 561 FREQUENCY VALUES ARE COMPRESSED TO:

```
01 0000 0000 0000 0002    SOS
08 8000 0000 0000 007F    BWX=0.5
08 C800 0000 0000 0081    XMIN=200.0
```

TWO WORDS OF COMPRESSED INFORMATION

```
01 1400 0A80 0140 0598
00 1EB6 100A 0191 0050
```

and at every 5 cms $^{-1}$ between 2,000 and 200 cms $^{-1}$. The lowest Limit of Significance (LSX) for the wave number and transmittance scales were taken as one cm $^{-1}$ and one percent respectively. Thus the regenerated information will be within 0.25 cm $^{-1}$ and 0.25 percent of the original wave number and transmittance values. If the number of data-points was increased (e.g., by an optical scan procedure), the saving would be even more substantial.

Normally, the information in Table II (or, if compression was not achieved, i.e., SOS(J) =-0, its uncompressed form) would be compressed by a recursive bit-pattern recognition technique in the ALPHANUMERIC COMPRESSOR [7], thus reducing fast (internal) storage requirements by another substantial factor. This highly compressed information could be reduced by another factor of at least 20 with the customary Binary-Compression routines [1] before it is stored in the external devices.

It should be noted that further savings in the internal storage requirements may be affected by modifying NUPAK so that the commands F and NS are not stored with every packed word. Another potential improvement is to extend the definitions of F and NS so that patterns of repeats located anywhere in the string will be identified. These modifications can be easily made for machines with accumulator and minor quotient registers.

Special thanks are due to Beckman Instruments Inc. at Silver Spring, Maryland for supplying and giving permission to reproduce their standard IR-12 spectrum of polystyrene film. Dr.A.L.Goldman, Mr.Paul Meissner and Dr. S. J. Tauber of the National Bureau of Standards are thanked for their critical reviews of the manuscript.

This work was supported by Transfer of Funds GN-455 from the National Science Foundation as a part of the Chemical Information Program jointly financed by the Department of Defense, the National Institutes of Health, and the National Science Foundation.

REFERENCES

1. Hutchinson, H. P., "Computers and Chemistry. What One Fool Can Do, Another Can." Chemistry in Britain 2, 396-400 (1966).

2a. Evans, D. C., "Computer Logic and Memory," Scientific American 215, 74-85 (1966).

b. McCarthy, J., "Information," ibid., 64-73. See also other articles in this issue devoted to information and its processing by computers.

3. Massey, H. N. and W. E. Smith, "Implementation Investigations of ZVA/FZA Compression Algorithms," Sunnyvale, Calif., Lockheed Missiles and Space Co. Rept. No. LMSC-669224, February 1966, 156 p. (NASA CR-76389).

4. University of Illinois, Urbana, The PREST (IBM), BININ, and BINOUT routines developed for their IBM 7094/1401 configuration.

5. deMaine, P. A. D. and G. K. Springer, "An Automatic Curve-Fitting Method" (in preparation); Appendix A contains a description of the BIN Procedure.

6. deMaine, P. A. D. and R. D. Seawright, "The Self-Judgment Principle in Scientific Data Processing," Ind. Eng. Chem. 55, 29-32 (1963); P. A. D. deMaine, "The Self-Judgment Methods of Curve Fitting," Comm. Assoc. Computing Machinery 8, 518-526 (1965); P. A. D. deMaine and R. D. Seawright, "Digital Computer Programs for Physical Chemistry," MacMillan Co., New York, Vol. I, 1963, 423 p.; Vol. II 1965, 493 p.

7. Marron, B. A. and P. A. D. deMaine, "Automatic Data Compression" (in preparation).

8. deMaine, P. A. D. and B. A. Marron, "The SOLID SYSTEM I. A Method for Organizing and Searching Files," in G. Schecter [Ed], "Information Retrieval. A Critical Review." (Proceedings of the Third Annual National Colloquium on Information Retrieval, held in May 1966) Thompson Book Co., Washington, D. C., 1967, 282 p.

9. Laderman, J. and M. Abramowitz, "Application of Machines to Differencing of Tables," J. Am. Statistical Assn. 41, 233-237 (1946).

Part II

THE SOLID SYSTEM. III. ALPHANUMERIC COMPRESSION

P. A. D. deMaine
B. A. Marron
K. Kloss

An algorithm for compressing alphanumeric information is described. Unlike other methods which depend upon frequency of occurrence of words in a particular class of publications, this scheme is language and content independent since the information for compression is obtained from the text itself. The compressed bit stream is preceded by sufficient information for automatic reconstruction of the original bit stream whenever the system requires it. Even with this additional information required for expansion, compression rates approaching 40% have been achieved. Because this ALPHANUMERIC COMPRESSOR (ANPAK) is fully automatic and self-organizing, it can operate on information which has already been compressed via the NUMERIC COMPRESSOR (NUPAK).

Key Words:
alphanumeric compression, information handling, high-speed information transmission, information storage and retrieval, systems analysis.

1. INTRODUCTION

It is envisaged that in the Self-Organizing Large Information Dissemination System (SOLID SYSTEM) [1] the fully automatic COMBINED COMPRESSOR (COPAK) will be used to reduce greatly the slow (external) storage required and to increase the "rate of information transmission" through the computer. COPAK will also automatically decode the compressed information on an item-by-item basis when it is required. The three component compressors of COPAK (NUPAK, ANPAK and IOPAK) can be separately used to accomplish each of their respective tasks as outlined below. The State of System Commands (SOS), which contain two items (NAP and NDR), determine which compressor components are to be used in a particular storage or retrieval act. (See Figure 1.)

The NUMERIC COMPRESSOR (NUPAK) has already been described in Part I. It achieves substantial savings in both fast and slow storage by storing the numerical data in a highly compact form after its conversion to fixed-point format. In NUPAK, the lowest "Limit of Significance" (LS) of the original data is respected during compression,

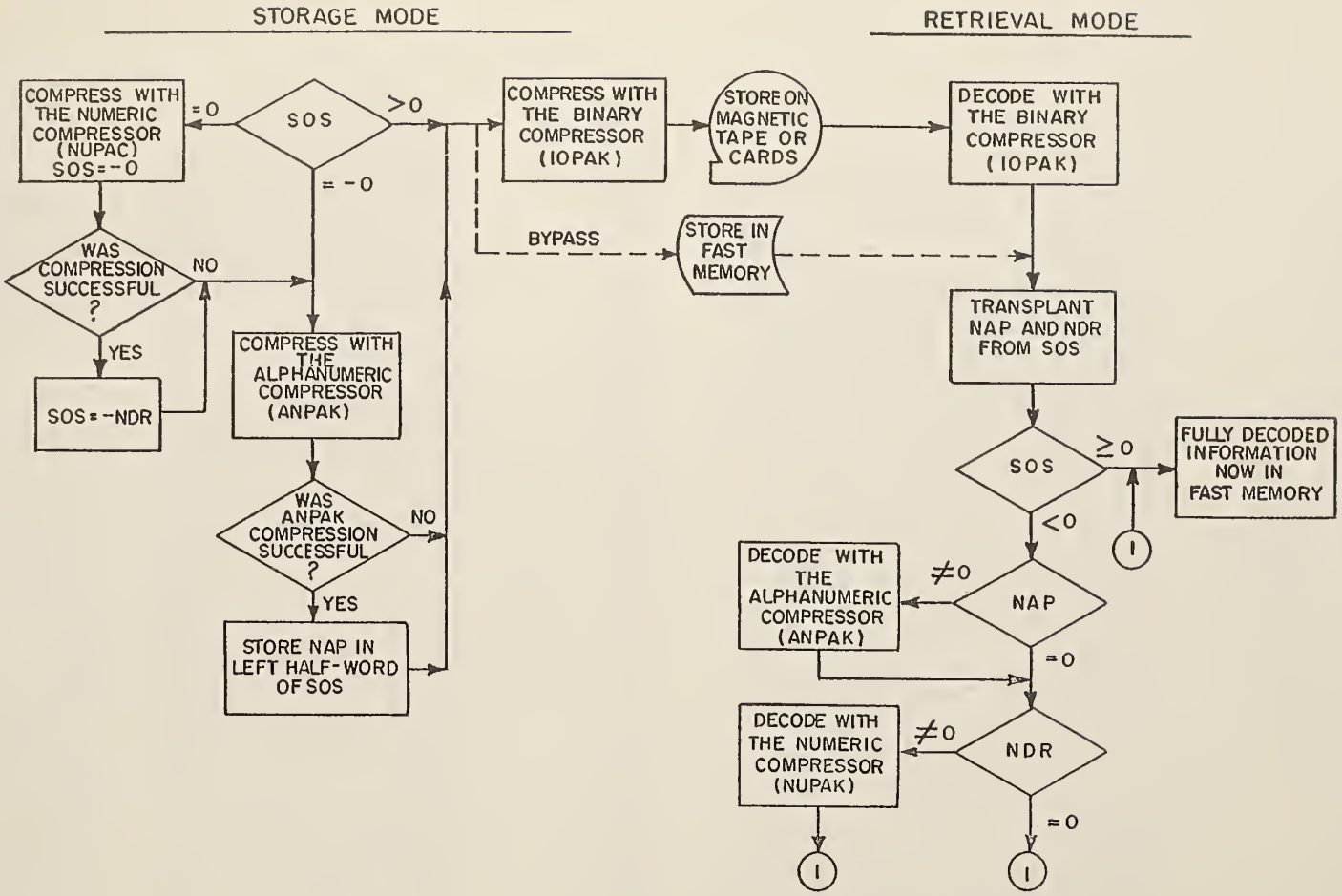


Figure 1. Schematic Flow-Chart showing the interrelationships between the three components (NUPAK, ANPAK and IOPAK) of the numeric-alphanumeric-binary compressor (COPAK) in the storage (encoding) and retrieval (decoding) modes. NAP and NDR are computed in the storage mode. Note that IOPAK is operational only for slow storage (magnetic tape, cards).

so that these data can be regenerated to well within this limit.

The ALPHANUMERIC COMPRESSOR (ANPAK) described here achieves savings in both fast and slow storage by an automatic, recursive bit-pattern recognition technique. There is no loss of information with ANPAK.

The IN-OUT COMPRESSOR (IOPAK), which will be closely patterned on the University of Illinois BININ and BINOUT routines [3], is to store the card image of binary coded information. No savings in fast memory (i.e., primary and disk storage) could be effected with IOPAK.

With COPAK, information compressed by NUPAK is automatically processed by ANPAK to obtain additional savings in both fast and slow storage. (See Figure 1.) Compressed information from ANPAK (or if no compression occurred, either the compressed information from NUPAK or the original data) is automatically processed by IOPAK before it is stored on tape or cards. IOPAK is inoperative if the information is to be stored in fast memory.

In the decoding procedure information entering the system from slow storage (i.e., tapes or cards) is automatically decompressed by the decoder component of IOPAK.(See Figure 1.) Next, the State of System Commands, constructed by the computer in the compression stages of NUPAK and ANPAK, are used to determine the sequential decompression steps to be taken by the decoder components of ANPAK and NUPAK. The decoding (or decompression) procedure is fully automatic.

The ALPHANUMERIC COMPRESSOR (ANPAK) and its interrelationship with the NUMERIC COMPRESSOR (NUPAK) described in Part I of this Technical Note, are discussed in the following sections.

2. PRINCIPLES OF THE ALPHANUMERIC COMPRESSOR (ANPAK)

In the SOLID SYSTEM, input to ANPAK is either comressed numerical information from NUPAK or the original input information if compression via NUPAK failed or was not called upon. It should be noted that ANPAK is a recursive bit-pattern recognition technique and is therefore linguistically independent, i.e., it is equally applicable to Russian text or English text, to eight-bit ASCII representation or a six-bit BCD character code. Further, because the original bit-pattern can be regenerated, no loss of information results from using ANPAK.

A. DEFINITIONS

For illustrative purposes, it is supposed that a string of JI machine words, each with N1 bits, is to be compressed. Here the string will be considered a single word (T) with N2 (=JI·N1) bits. The following definitions are associated exclusively with the ALPHANUMERIC COMPRESSOR (ANPAK).

A Code contains 2^{CW} code-words, each with CW bits. N1/CW must be a positive integer. Thus T can be regarded as a sequence of code-words.

A Lexicon (TL) discloses which of the 2^{CW} code-words have been used to achieve compression and in what manner.

A Cord (CD) contains R code-words consecutive in the string T. N3 (=R.CW), the number of bits in the cord, cannot exceed N1; R is a positive integer.

The Bit-Map (BM) of one of the 2^{CW} code-words discloses the positions of that code-word in the string T. Terminal zeros in a bit-map are omitted, e.g., for T=101|011|010|101|010|100|

010|000 the bit-map of 101 is 1001, meaning that 101 is the first and fourth (and only these) of the successive code-words of length CW in T. (Note: Bit-maps are used only in Type II Compression.)

In Type I Compression, an unused code-word is substituted for a cord.

In Type II Compression, code words are removed from the string, and their locations are designated by bit-maps.

Details of both types of compression are given in section D.

A string is irreducible if compression cannot be achieved.

B. COMPRESSION PROCEDURE

Simple illustrations of the compression procedure are given in Section F.

Step 1: The smallest value of CW is computed from N1 and the input information. For numeric information the initial value of CW is the smallest number greater than four which divides N1 exactly. For alphanumeric information CW is set equal to six or eight (for example, BCD or ASCII code).

Step II: The lexicon (TL) associated with the CW bit code is constructed as follows. An array Y is constructed which consists of 2^{CW} consecutive machine words, initially set equal to zero, corresponding in a definite order to the 2^{CW} possible CW bit binary words. The code-words of string T are examined, and the Ith in Y is used as an indicator of the presence of the Ith

binary word (in the specified ordering) as a code-word in T. Then the zero-words remaining in array Y are tallied in NRL, and the corresponding unused code-words are stored in the array TL.

Step III: The value of R is set to its maximum. (In our programs the maximum value of R is 9.) The search begins with the longest cord, i.e., maximum R so that shorter cords which are contained in the long cord are not replaced by a code-word first. If this did occur, the savings achieved would be smaller. However, it is realized that this somewhat arbitrary choice of beginning with maximum R may result in less savings in certain cases. Further information is needed.

Step IV: NR, a counter, is set equal to zero.

Step Va: If NRL $\neq$ 0, Step Vb is executed. For NRL = 0, both R and NRL are set equal to one, and Step Vb is executed.

Vb: The N3-bit cord, CD_{N3} (where N3=R•CW), is set equal to bits (NR•CW + 1) to (NR•CW + N3) in string T.

Vc: A search of string T with CD_{N3} discloses whether or not a compression can be achieved. (The criterion for successful compression is that the number of bits which can be removed from the string must be greater than the number of bits which must be added to the string to permit automatic decompression.) In this searching procedure, if there is a match between CD_{N3} and the N3 bit cord in the string, the next attempted match will

be with a cord in the string beginning N3 bits (the cord length) further along. If a mismatch occurs, the next attempted match will be with a cord in the string beginning CW bits (the code-word length) further along. If R>1, compression is achieved by substituting the first unused code-word in TL for CD_{N3} wherever it occurs (Type I Compression). For R=1, a bit-map for CD_{N3} (here the code-word) is constructed and the string is compressed by removing the cord wherever it is found and NRL is decreased by one (Type II Compression). In our present programs, Type II Compression is confined to the first 32 code-words of the string. If a saving is achieved, a composite code-word (CCW_i) in the array TL is constructed in one of the following forms:

Type I (Code-word substituted for cord) (R > 1)

Code Word (CW-bits)	R (four bits)	Cord (N3 bits)

Type II (Bit-Map of Code Word) (R = 1)

Code Word (CW bits)	R (four bits)	No. bits in Bit-Map (NB) (five bits)	Bit-Map (NB-bits)

Vd: If compression was achieved, the above procedure beginning with Step IV is repeated with the compressed string. If no compression was achieved NR is incremented by one and control gets to Step V. If all N3-bit cords (CD_{N3}) have been examined, control goes to Step VI.

Step VI: R is decreased by one. If $R \geq 1$, control goes to Step IV; for R = 0, control goes to Step VII.

Step VII: If compression was achieved, control goes to Step VIII for the new string assembly. If no compression was achieved, CW is incremented by steps of one until N1/CW is again an integer. If N1=CW, the compression is complete and control goes to the calling system. Otherwise, control goes to Step II, where the lexicon associated with the new code is constructed.

Step VIII: The irreducible string (T_i) and its associated lexicon (TL) are combined in a compact self-defining string (I) thus:

BJI_i	T_i	ND_i	CW_i	TL_{1i}	TL_{2i}	...	...	TL_{ri}	...	...

... (I)

Here BJI_i is the number of bits in the irreducible string (T_i). ND_i is the number of COMPOSITE words in the lexicon for the code with CW_i bits; these (TL_{1i}, TL_{2i}, etc.) are arranged in the reverse order from that in which they were constructed. NAP (the number of successful compressions with different strings like I) equals i. The new string I is processed, beginning with Step II, with the value of CW unaltered.

This procedure (with newly defined strings) is repeated until no further saving can be achieved. (See Step VII.) The final form of the compressed information consists of a single string like I plus one word (NAP). NAP is stored in the State of System Command (see section E).

C. DECOMPRESSION PROCEDURE

To regenerate the original string T the following procedure is executed:

Step I: If NAP =0, no compression was achieved and control returns to the calling system; otherwise it goes to Step II.

Step II: The string T_i, with i = NAP, is expanded by using the ND_i composite words consecutively in the reverse order from that in which they were constructed. This means that I is first split into its components BJI_i, T_i, and ND_i, CW_i, TL_{1i}, TL_{2i}, ...; then T_i is expanded to T_i' with TL_{1i}. Next T_i' is expanded, in turn, with TL_{2i} and so on. This procedure is repeated until the lexicon associated with the CW_i-bit code has been used.

Step III: NAP is decreased by one, and if NAP $\neq$ 0, control goes to Step II, with T_{i-1} in place of T_i.

D. STRUCTURE OF COMPRESSED INFORMATION

In addition to NAP, which is stored in SOS (see section E), the compressed information consists of a single compact self-defined string, like I, with a mixture of fixed and variable fields. The lexicon of composite code-words (TL_{ji}), associated with the code with CW_i bits and NAP = i, also contains fixed and variable field information thus:

Type I Compression ($R_{ji} \neq 1$)

ACW	R_{ji}	CD_{ji}

Here ACW_{ji} is the jth code-word associated with the CW_i-bit code and NAP = i. CD_{ji} is the cord which was replaced by ACW_{ji}; R_{ji} is the number of code words in cord CD_{ji}.

Type II Compression ($R_{ji} = 1$)

ACW_{ji}	$R_{ji}=1$	NB_{ji}	BM_{ji}

Here BM_{ji} is the bit-map associated with the code-word ACW_{ji}. NB_{ji} indicates the number of bits in the bit-map (BM_{ji}), which has no terminal zeros. In this type of compression the code-word (ACW_{ji}) is removed from the string if it occurs anywhere in the first 32 code-words. This limit (32 code-words) is incorporated in our programs to permit the use of single machine words (65 bits) for each composite code-word in the lexicon. The bit-map actually defines the locations in the original string of the code-word ACW_{ji}.

The fixed fields in (I), (BJI_i, ND_i, CW_i, R_{ji} and NB_{ji}), are defined thus:

BJI_i (18 bits) is the number of bits in the string T_i.

ND_i (5 bits) is the number of composite code-words in lexicon TL_{ji} associated with the CW_i-bit code.

CW_i (5 bits) is the number of bits in the code associated with NAP = i.

R_{ji} (4 bits) is the number of code-words in the associated cord (CD_{ji}).

NB_{ji} (5 bits) indicates the number of bits in the bit-map (BM_{ji})

if R_{ji} = 1 and type II compression was achieved.

The variable fields (T_i, ACW_{ji} and BM_{ji}) are defined next:

T_i is the irreducible string obtained by compressing the string which precedes I. This may have been the original string (i=1) or may itself have been constructed from an irreducible string and its lexicon (i>1).

ACW_{ji} is the jth code-word associated with the CW_i bit-code.

CD_{ji} is the cord associated with ACW_{ji}

BM_{ji} is the bit-map associated with the code-word ACW_{ji}.

E. STATE OF SYSTEM COMMANDS (SOS, LJ, MODE)

The values for SOS and MODE together define the status of the information in the system. In the storage mode (MODE = 1), the value of SOS is changed after executing alphanumeric compression (ANPAK). In the retrieval mode (MODE = 0) the redefined value of SOS, which is stored together with the compressed or expanded string in the file, determines the decoding procedures to be used.

Storage Command (SOS)

SOS < 0 means that numeric information obtained as output from the NUMERIC COMPRESSOR (NUPAK) is to be compressed further by the ALPHANUMERIC COMPRESSOR (ANPAK) if possible.

SOS =-0 means that alphanumeric information must, if possible, be compressed with ANPAK.

SOS > 0 means that the information is not to be compressed with NUPAK or ANPAK. The string of information is stored in expanded form in the file.

If SOS < 0, the minimum value of CW in Step I is set equal to its least value greater than four which exactly divides N1 (number of bits per

machine-word). If SOS = 0, the minimus value of CW equals six or eight (BDC or ASCII Code).

Retrieval Command (SOS)

In the storage mode (MODE = 1) after executing the ANPAK compression, the value of NAP is inserted in the left half-word of SOS. This redefined value is stored together with the string of information (compressed (NAP $\neq$ 0), expanded (NAP = 0)) in the file. In retrieval operations, if SOS < 0 and NAP = 0, control passes to the decoder part of NUPAK (refer to Part I of this Technical Note). If SOS > 0, control goes to the search procedure for the next instruction.

The post-operation command (LJ) enters the system, together with SOS and the original string of information (T), as input. After the command SOS is executed (see above), control passes to LJ, which determines the next operation thus:

LJ<0(STOP); LJ>0(CALLING PROCEDURE); LJ=0(READ FILE).

For instance, if LJ > 0 the next batch of input is a retrieval problem and control goes to the CALLING PROCEDURE. In the SOLID SYSTEM [1] the JOB-LIST is used as a modifier of LJ.

F. ILLUSTRATIONS OF COMPRESSION METHODS

To illustrate the two kinds of compressions possible with ANPAK, the following fragments of information actually compressed with the NBS Pilot Data Processor are given. (In Section 3 the results of compressing a draft of this paper are discussed.)

1. Type I Compression (Substituted Code-Word)

Suppose that the following sentence is to be compressed via the ALPHANUMERIC COMPRESSOR (ANPAK)--CW was set equal to six since input was in BCD format:

HERE IS DESCRIBED THE FULLY AUTOMATIC ALPHANUMERIC COMPRESSOR FOR USE EITHER AS A SECOND STAGE OR AS AN ALTERNATIVE TO THE NUMERIC COMPRESSOR.

The irreducible string T_1 obtained with the Pilot via ANPAK is:

HEReISΔDESCRIBEDdFULLYΔAUTOMATICΔALPHAbΔFORΔUSeEITHEcΔSECONDΔ

STAGeOcNΔALTERNATIVeTOdbR.

(Δ signifies a blank space, and the lower case letters are used in place of the numeric codes normally obtained with a computer.) The lexicon of composite code-words is as follows:

Code-Word	R	Cord
a	9	NUMERICΔC
b	9	aOMPRESSO
c	6	RΔASΔA
d	5	ΔTHEΔ
e	2	EΔ

The irreducible string and its self-defining lexicon are assembled in the final step of the compression procedure.

In the fully automatic decoding procedure ANPAK first separates the components and, beginning with the last defined code-word, substitutes the cords (Type I) or code-word (Type II) back into the string.

2. Type I and Type II Compression:

The following arbitrary string of 44 characters (six bit code) was used:

LMKXKLLLKTKKLPQ*,KKTTLLLKKKTQSPP::P*QSQWEZTY

The lexicon for Type I compression is:

Code-word	R	CORD
A	4	LLLR

The lexicon for Type II compression is:

Code-word	R	NB_{11}	Bit-Map (BM_1)
K	1	20	00101001100000110011

(NB_{11} + 1) is the number of bits in the Bit-Map (BM_1). BM_1 is constructed after Type I Compression occurs. The irreducible string T_1 is

LMXATLPQ*,TTATQSPP::P*QSQWEZTY

Note that the Type II compression in our program applies to the first 32 code-words in each string. In this example, further Type II compression, e.g., for the code-word T, cannot be achieved.

3. USE OF ANPAK AND COPAK

For illustrative purposes the first four pages of typed material for an earlier draft of this paper were compressed with ANPAK. The original string of BCD characters consisted of 15,180 bits. The compressed string, including its lexicon and commands for decompression (via ANPAK decoder), consisted of 9,291 bits. Thus even in this short example there is a saving of 39 percent in high-speed memory storage.

The program now in existence permits the predefinition of any lexicon simply by adding any desired bit-configuration to the head of the input string. This allows the user to speed up the algorithm by indicating certain substrings as obvious candidates for compression.

It is important to note that while ANPAK utilizes the natural frequency of alphanumeric information it is independent of the language or

coding of the information; because of this fact numeric information compressed via NUPAK is automatically compressed via ANPAK in the second stage of COPAK. (See Figure 1.)

The alphanumeric compressor with its fully automatic decoder (ANPAK) described here has been coded for the experimental Pilot Data Processor. However, ANPAK can be coded for any computer with registers and minimal Boolean algebra capabilities.

ACKNOWLEDGEMENT

This work was supported by Transfer of Funds GN-455 from the National Science Foundation as part of the Chemical Information Program supported jointly by the Department of Defense, the National Institutes of Health, and the National Science Foundation.

REFERENCES

1. deMaine, P. A. D., and B. A. Marron, "The SOLID SYSTEM I. A Method for Organizing and Searching Files," in G. Schechter [Ed], "Information Retrieval. A Critical Review." (Proceedings of the Third Annual National Colloquium on Information Retrieval, held in May 1966) Thompson Book Co., Washington, D. C., 1967, 282 p.

2. University of Illinois, Urbana, The PREST (IBM), BININ, and BINOUT routines developed for their IBM 7094/1401 configuration.

* U.S. GOVERNMENT PRINTING OFFICE: 1967—301-017/4

THE NATIONAL BUREAU OF STANDARDS

The National Bureau of Standards[1] provides measurement and technical information services essential to the efficiency and effectiveness of the work of the Nation's scientists and engineers. The Bureau serves also as a focal point in the Federal Government for assuring maximum application of the physical and engineering sciences to the advancement of technology in industry and commerce. To accomplish this mission, the Bureau is organized into three institutes covering broad program areas of research and services:

THE INSTITUTE FOR BASIC STANDARDS . . . provides the central basis within the United States for a complete and consistent system of physical measurements, coordinates that system with the measurement systems of other nations, and furnishes essential services leading to accurate and uniform physical measurements throughout the Nation's scientific community, industry, and commerce. This Institute comprises a series of divisions, each serving a classical subject matter area:

—Applied Mathematics—Electricity—Metrology—Mechanics—Heat—Atomic Physics—Physical Chemistry—Radiation Physics—Laboratory Astrophysics[2]—Radio Standards Laboratory,[2] which includes Radio Standards Physics and Radio Standards Engineering—Office of Standard Reference Data.

THE INSTITUTE FOR MATERIALS RESEARCH . . . conducts materials research and provides associated materials services including mainly reference materials and data on the properties of materials. Beyond its direct interest to the Nation's scientists and engineers, this Institute yields services which are essential to the advancement of technology in industry and commerce. This Institute is organized primarily by technical fields:

—Analytical Chemistry—Metallurgy—Reactor Radiations—Polymers—Inorganic Materials—Cryogenics[2]—Office of Standard Reference Materials.

THE INSTITUTE FOR APPLIED TECHNOLOGY . . . provides technical services to promote the use of available technology and to facilitate technological innovation in industry and government. The principal elements of this Institute are:

—Building Research—Electronic Instrumentation—Technical Analysis—Center for Computer Sciences and Technology—Textile and Apparel Technology Center—Office of Weights and Measures —Office of Engineering Standards Services—Office of Invention and Innovation—Office of Vehicle Systems Research—Clearinghouse for Federal Scientific and Technical Information[3]—Materials Evaluation Laboratory—NBS/GSA Testing Laboratory.

---

[1] Headquarters and Laboratories at Gaithersburg, Maryland, unless otherwise noted; mailing address Washington, D. C., 20234.

[2] Located at Boulder, Colorado, 80302.

[3] Located at 5285 Port Royal Road, Springfield, Virginia 22151.