[*Mechanical Translation*, vol.5, no.1, July 1958; pp. 2-7]

An Input Device for the Harvard Automatic Dictionary†

Anthony G. Oettinger, Computation Laboratory,
Harvard University, Cambridge, Massachusetts

A standard input device has been adapted to permit transcription of either Roman or Cyrillic characters, or a mixture of both, directly onto magnetic tape. The modified unit produces hard copy suitable for proofreading, and records information in a coding system well adapted to processing by a central computer. The coding system and the necessary physical modifications are both described. The design criteria used apply to any automatic information-processing system, although specific details are given with reference to the Univac I. The modified device is performing satisfactorily in the compilation and experimental operation of the Harvard Automatic Dictionary.

THE PROPERTIES of a given automatic information-processing machine depend primarily on the algorithms the machine is capable of applying to the tokens [1] for the abstract elements it is said to process. Configurations of the states of sets of two-state devices, or pulse trains where pulses are present or absent in definite time intervals, are commonly used as tokens in contemporary machines. Abstract elements, e.g., the integers, are named by symbols of various kinds. For example, the numerals "2", "II", and "10" all name the number 2. Likewise, various symbols can be used to name tokens. It is a useful and widely accepted convention to use the symbol "0" as the name for one state of a two-state device, and the symbol "1" as a name for its other state. Frequently, the symbols "0" and "1" are used also as binary numerals. In a context where both these usages occur, a string such as "1001" functions homographically both as a name for the number 9 and as a name for a particular configuration of a set of four two-state devices. This practice is confusing in discourse about machines intended for or adapted to purposes other than numerical computation, especially when the relation between machine tokens and abstract elements is the chief subject of discussion. In this paper, therefore, "0" and "1" will be used exclusively as the names of tokens.

The mapping between machine tokens and the abstract elements a given machine is said to process can be regarded as defined by the input and output hardware of the machine. For example, if a pulse train 1010100 is to be regarded as a token for the letter A, it is desirable to arrange matters so that such a pulse train will cause a printer to print the literal "A". When an order relation exists among the tokens in a machine, as imposed, for example, by comparison and branch instructions, and when the abstract elements themselves are an ordered set, it is usually desirable to relate abstract elements and tokens by an order-preserving mapping. For example, in a machine designed to recognize 1010100 to be "smaller" than 0010101 and 0010101 in turn to be smaller than 0010110, the mapping A — 1010100, B — 0010101, C — 0010110 preserves normal alphabetic order, whereas A — 0010101, B — 1010100, C — 0010110 does not.

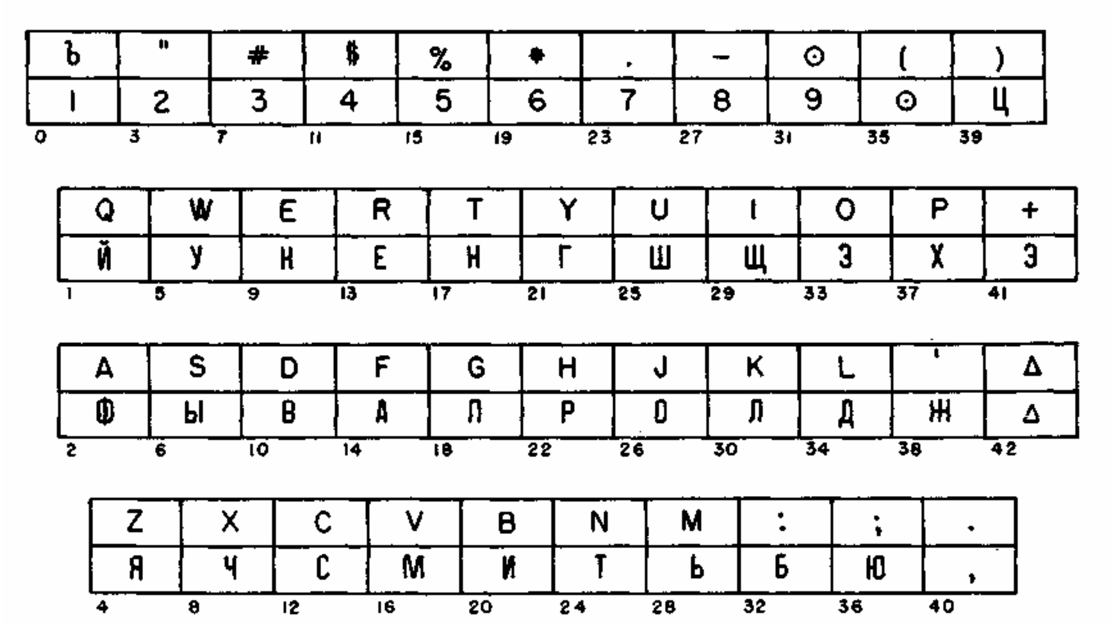
---

† This work has been supported in part by the Harvard Foundation for Advanced Study and Research, the United States Air Force, and the National Science Foundation.

1. This term was originated by C. S. Peirce. For an explanation of the underlying distinctions, see H. Reichenbach, Elements of Symbolic Logic, Macmillan, New York, 1947, p.4.

The Univac I computer is currently in use at the Harvard Computation Laboratory in connection with the development of an operating automatic dictionary[2] and for basic research on the problems of automatic translation from Russian into English. The normal mapping between numbers, letters of the Roman alphabet, punctuation marks, and other standard symbols on the one hand, and machine tokens on the other, is given in Figure 2 by the columns headed "Upper Case" and "Binary Code" (except for key no. 0). This mapping is established by all input and output devices associated with the machine, in particular by the Unityper, which is used to record information onto magnetic tape, and by the High-Speed Printer, which is the major output unit. Thus, when an A is typed, a token 1010100 is recorded, and such a token will in turn cause the High-Speed Printer to print an A.

Adapting a machine like the Univac to handle Cyrillic letters is conceptually a trivial matter. To permit alphabetization of Cyrillic material, an order-preserving mapping between the Cyrillic alphabet and Univac tokens is necessary. Many such mappings can readily be established. Once this has been done, the internal operation of the machine with Cyrillic material presents no difficulties. However, unless the input and output devices are physically altered, certain practical problems obviously arise.



Keyboard Layout

Figure 1

2. Oettinger, A. G., Foust, W., Giuliano, V., Magassy, K., Matejka, L., "Linguistic and Machine Methods for Compiling and Updating the Harvard Automatic Dictionary" (To be presented at the International Conference on Scientific Information, Washington D.C., November 1958, and published in the Proceedings of the conference).

As a first step, it is simple to cover the keys on the Unityper with keytops labelled with Cyrillic letters. From the point of view of typing ease and accuracy the most desirable keyboard layout (Fig. 1) is one in standard use on ordinary Cyrillic typewriters. Unfortunately, merely replacing keytops solves only a part of the practical problem. First, the typewriter

KEY #	LOWER CASE	UPPER CASE	BINARY CODING 1	2	3	4	5	6	7
0	(I)	(S) ь	0/1	0/1	0/1	0/0	1/1	0/0	0/1
1	Й (;)	(Q)	0/1	0/1	1/0	0/1	0/0	1/1	1/1
2	Ф (M)	(A)	1/1	1/0	0/1	0/0	1/1	1/0	1/0
3	(2)	(")	1/1	0/1	0/0	0/0	1/0	0/0	1/1
4	Я (Y)	(Z)	0/1	1/1	1/1	1/1	0/1	1/0	1/0
5	У (L)	(W)	0/1	1/1	0/1	0/1	1/0	1/0	0/1
6	Ы (T)	(S)	1/1	1/1	1/1	0/0	1/1	1/0	0/1
7	(3)	(#)	1/1	0/0	0/1	0/1	1/1	1/0	0/1
8	Ч (Q)	(X)	1/1	1/1	0/1	1/1	0/0	1/1	1/0
9	К (B)	(E)	0/1	0/0	1/1	0/1	1/0	0/0	1/0
10	В (3)	(D)	1/1	0/0	0/1	0/0	1/1	1/1	0/1
11	(4)	($)	0/1	0/1	0/0	0/1	1/1	1/0	1/1
12	С ())	(C)	0/0	1/0	0/1	0/0	0/1	1/1	1/0
13	Е (6)	(R)	1/0	0/1	0/0	1/1	0/1	0/0	1/0
14	А (1)	(F)	0/0	0/0	0/1	0/1	1/0	0/0	0/1
15	(5)	(%)	0/0	0/1	0/1	1/1	0/1	0/0	0/1
16	М (D)	(V)	1/0	0/1	1/1	0/1	1/0	1/0	1/0
17	Н (E)	(T)	1/1	0/1	1/1	1/0	0/1	0/1	0/0
18	П (H)	(G)	1/0	0/0	1/1	1/1	0/0	1/1	1/0
19	(6)	(*)	1/1	0/1	0/0	1/1	0/1	0/1	1/0
20	И (9)	(B)	1/0	0/0	0/1	1/0	1/1	0/0	0/1
21	Г (4)	(Y)	0/0	0/1	0/1	0/1	1/0	1/1	1/1
22	Р (I)	(H)	0/1	0/0	1/1	1/1	1/0	0/1	0/1
23	(7)	(·)	1/1	0/0	0/1	1/0	0/0	1/1	0/0
24	Т (K)	(N)	0/1	1/1	0/0	0/1	1/0	0/0	1/0
25	Ш (R)	(U)	0/0	1/1	0/1	1/0	1/1	0/1	0/1
26	О (G)	(J)	0/1	0/1	1/0	1/0	0/1	1/0	0/0
27	(8)	(-)	0/0	0/0	0/0	1/0	0/0	1/1	1/0
28	Ъ (U)	(M)	0/1	1/1	1/0	0/0	1/1	1/1	1/1
29	Щ (+)	(I)	1/0	1/0	1/1	0/1	0/1	1/0	1/0
30	Л (C)	(K)	0/0	0/1	1/0	0/0	1/1	1/0	0/1
31	(9)	(8)	1/1	0/0	0/0	1/0	1/0	0/1	0/1
32	Б (2)	(:)	1/0	0/1	0/1	0/0	1/0	0/1	1/0
33	З (8)	(0)	0/0	0/1	0/0	1/1	0/0	1/0	1/1
34	Д (5)	(L)	0/0	0/1	0/0	1/0	0/1	0/1	0/0
35	(0)	(()	1/1	0/0	0/0	0/1	0/1	1/1	1/1
36	Ю (X)	(;)	1/0	1/0	1/1	1/0	0/0	1/1	0/1
37	Х (N)	(P)	1/0	1/1	0/0	1/1	0/0	0/1	0/0
38	Ж (7)	(')	1/0	0/0	0/0	1/1	0/1	1/0	0/1
39	Ц (P)	())	0/0	1/1	0/0	1/0	0/0	1/1	0/1
40	(,)	(.)	1/1	0/0	1/1	0/0	0/0	0/1	1/0
41	Э (V)	(+)	0/1	1/1	1/1	1/0	0/0	0/1	0/1
42	(Δ)	(Δ)	0/0	0/0	0/0	0/0	0/0	0/0	1/1
43	TRIP		0/0	0/0	0/0	0/0	0/0	0/0	0/0

NOTES:

1) 0= NOTCH CUT, 1= NO CUT

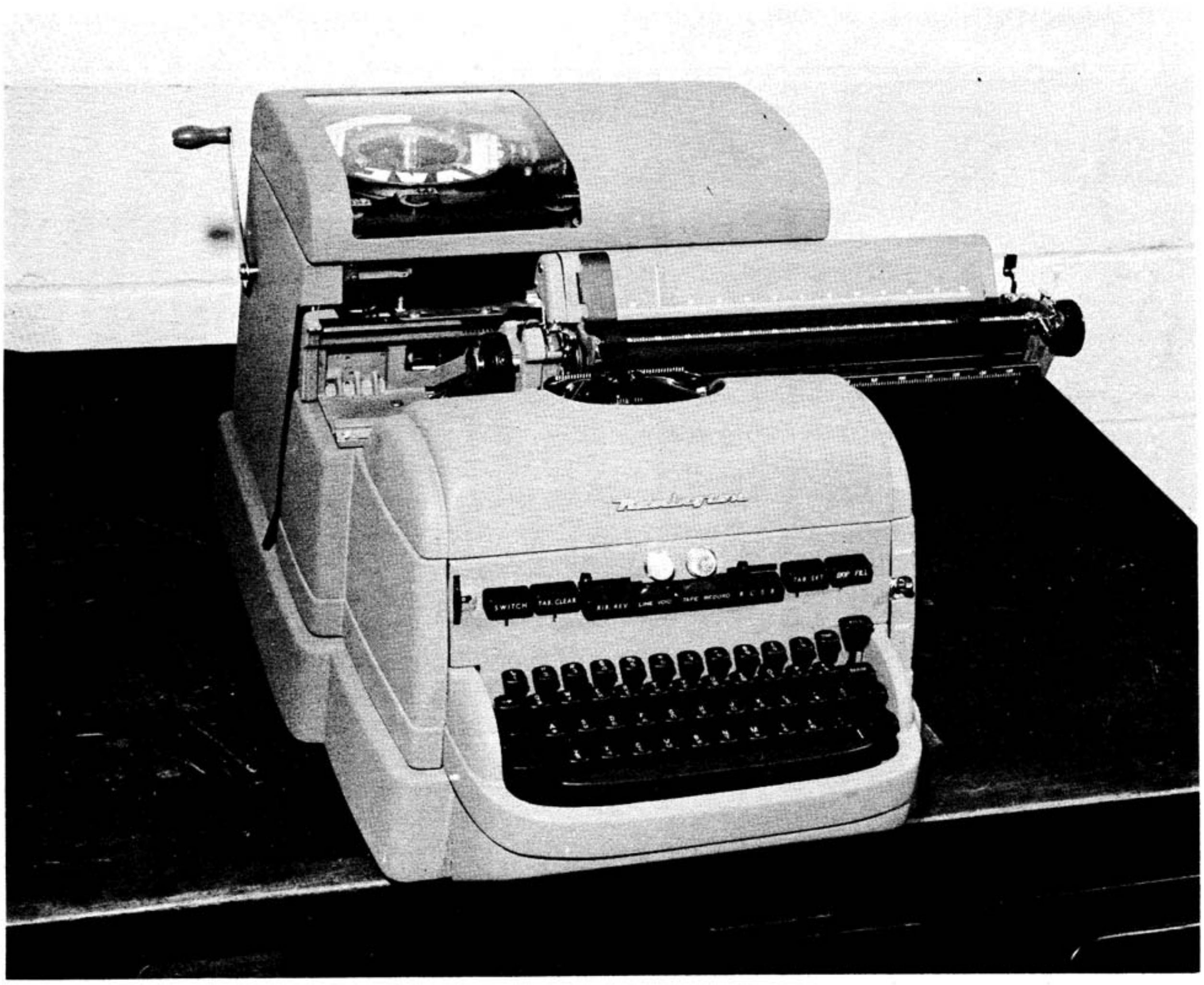
2) WHEN THE CHARACTER PRINTED ON THE HARD COPY IS DIFFERENT FROM THAT RECORDED ON MAGNETIC TAPE, THE FORMER IS SHOWN TO THE SIDE OF THE APPROPRIATE CIRCLE.

Definition of Mappings

Figure 2

continues to print Roman letters (e.g., Q for Й), a cryptographic transformation that makes proofreading most difficult. Second, the correspondence between the Cyrillic alphabet and machine tokens established in this way does not preserve Cyrillic alphabetic order. To reconcile these conflicting demands, a composition of two successive mappings can be used.[3] The first, established by the input device with covered keytops, leads to the representation of Cyrillic information in a "typewriter code." A subsequent code conversion is made automatically on the computer, at the expense of some running time, leading to the representation of Cyrillic letters in a "ranked code." The resultant mapping is order-preserving. In Figure 2, the Cyrillic letters are named in the "Lower Case" column. The token corresponding to a particular Cyrillic letter in the ranked code is named in the "Binary Coding" column, in the same row as the letter. The choice of this particular mapping was made for technical reasons

3. Ibid.



Modified Roman / Cyrillic Unityper

Figure 3

described in detail elsewhere.[4] Similar expedients have been used by others.[5]

Recently, we modified a standard Unityper to enable both the direct conversion from Cyrillic to ranked code, and the production of Cyrillic hard copy. The necessity for a costly intermediate code conversion by the computer itself is thereby eliminated, and proofreading is made relatively easy. The layout of the keyboard of the modified typewriter is shown in Figure 1. Figure 3 is a photograph of the actual machine. A sample of the hard copy produced by the modified Unityper is shown in Figure 4. The facility for interspersing standard and Cyrillic symbols is proving extremely useful in the recording of Russian texts, as illustrated in Figure 4.

---

4. Giuliano, V., "Programming an Automatic Dictionary" Design and Operation of Digital Calculating Machinery, Progress Report AF-49, Harvard Computation Laboratory, 1957, pp. I-42-I-45.

5. Edmundson, H.P., Hays, D.G., Renner, E.K., Button, R.I., "Manual for Keypunching Russian Scientific Text" RM-2061, RAND Corporation, 1957.

```
THE␣TEXT␣ON␣THIS␣PAGE␣HAS␣BEEN␣RECORDED␣ON␣MAGNETIC␣TAPE␣BY␣MEANS␣OF␣A␣MODIFIED␣UNITYPER.

TO␣DISTINGUISH␣SPACES␣FROM␣BLANK␣PAPER,␣THE␣SYMBOL␣"␣"␣IS␣PRINTED␣WHENEVER␣THE␣STANDARD␣SPACE␣BAR␣OR␣A␣SPECIAL␣SPACE␣KEY
␣IS␣STRUCK.

WHEN␣THIS␣UNITYPER␣IS␣SET␣FOR␣LOWER-CASE␣TYPING,␣CYRILLIC␣CHARACTERS␣ARE␣PRINTED␣AND␣RECORDED␣IN␣A␣SPECIAL␣CODE.␣␣ROMAN␣
CHARACTERS␣AND␣ARABIC␣NUMERALS␣ARE␣AVAILABLE␣IN␣THE␣UPPER-CASE␣SETTING.

THE␣AVAILABILITY␣OF␣ROMAN␣AND␣CYRILLIC␣CHARACTERS␣ON␣THE␣SAME␣KEYBOARD␣FACILITATES␣THE␣TREATMENT␣OF␣EQUATIONS␣AND␣OTHER␣
SPECIAL␣SYMBOLS␣OCCURRING␣IN␣TEXTS,␣E.G.:

     (1) РАССМАТРИВАЮТСЯ␣КОНЕЧНЫЕ␣МНОЖЕСТВА␣ОБЬЕКТОВ␣$A␣SUB␣1,...,␣A␣SUB␣N$␣,КОТОРЫЕ...
     (2) ...МОЖНО␣ЗАПИСАТЬ␣В␣ВИДЕ␣МАТРИЦЫ␣$EQUATION␣3$␣ГДЕ...

THE␣DOLLAR␣SIGNS␣IN␣THE␣EXAMPLES␣ARE␣USED␣AS␣BRACKETS␣WHICH␣SIGNAL␣THE␣TRANSLATING␣MACHINE␣TO␣TREAT␣THE␣SYMBOLS␣WITHIN␣B
RACKETS␣IN␣A␣SPECIAL␣WAY.

В␣ГЛ.␣␣$1$␣ЭТОЙ␣КНИГИ␣ДАНЫ␣НЕКОТОРЫЕ␣ВЕКТОРНЫЕ␣СООТНОШЕНИЯ␣И␣ВЫРАЖЕНИЯ␣*,*␣КОТОРЫЕ␣ПОЯСНЯЮТСЯ␣НИЖЕ␣*.*␣␣СКАЛЯРНОЕ␣ПРОИЗВЕ
ДЕНИЕ␣ДВУХ␣ВЕКТОРОВ␣$CAP.␣A␣BAR␣TIMES␣CAP␣B␣BAR$␣*.*␣␣ЭТО␣*--*␣СКАЛЯРНАЯ␣ВЕЛИЧИНА␣*.*␣␣В␣ПРЯМОУГОЛЬНЫХ␣КООРДИНАТАХ␣$EQUA
TION␣1$␣ГДЕ␣$A␣SUB␣X$␣*--*␣КОМПОНЕНТА␣ВЕКТОРА␣$A␣BAR$␣НА␣ОСЬ␣$X$␣*,*␣$B␣SUB␣X$␣*--*␣КОМПОНЕНТА␣ВЕКТОРА␣$B␣BAR$␣НА␣ОСЬ␣$X
$␣И␣Т.␣Д.␣␣ВЕКТОРНОЕ␣ПРОИЗВЕДЕНИЕ␣ДВУХ␣ВЕКТОРОВ␣ЕСТЬ␣ВЕКТОР␣*.*␣␣ВЕКТОРНОЕ␣ПРОИЗВЕДЕНИЕ␣ДВУХ␣ВЕКТОРОВ␣$A␣BAR$␣И␣$B␣BAR$␣
ЗАПИСЫВАЕТСЯ␣В␣ВИДЕ␣$A␣BAR␣TIMES␣B␣BAR$␣*.*␣␣ПРИ␣ИЗМЕНЕНИИ␣ПОРЯДКА␣ПЕРЕМНОЖЕНИЯ␣МЕНЯЕТСЯ␣ЗНАК␣ПРОИЗВЕДЕНИЯ␣*.*␣␣ТАКИМ␣ОБ
РАЗОМ␣*,*␣$EQUATION␣2$␣*.*␣␣В␣ПРЯМОУГОЛЬНЫХ␣КООРДИНАТАХ␣КОМПОНЕНТЫ␣ВЕКТОРНОГО␣ПРОИЗВЕДЕНИЯ␣РАВНЫ␣$EQUATION␣3$␣*.*␣␣

                              $END␣OF␣TEXT$
```

Demonstration Hard Copy Produced by the Modified Unityper

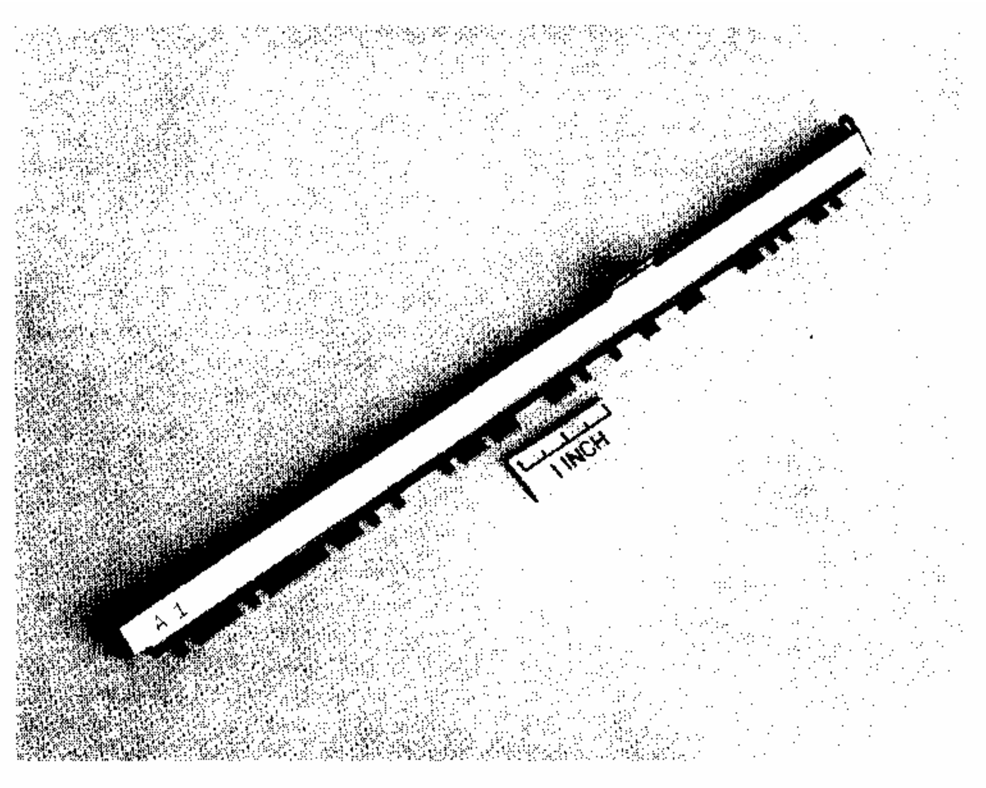
Figure 4

In lower case, the typewriter is Cyrillic. Except for three of the very low frequency letters, the layout is standard. In upper case, the typewriter functions as a standard model, except for the absence of a few special symbols normally available, and for the presence of one infrequently used Cyrillic letter. The mapping which obtains when the typewriter is in upper case is described by the "Upper Case" and "Binary Coding" columns of Figure 2. For example, 1101011 is a token for the letter Q. In lower case, the mapping is that described by the "Lower Case" and "Binary Coding" columns. For example, 0010011 is defined as a token for the Cyrillic letter Й.

The symbols circled in the "Lower Case" column are the normal correspondents of the tokens. For example, while 0010011 is defined as a token for Й in the ranked code, it is normally a token for the semi-colon. Therefore, since the output equipment has not been modified, Cyrillic material in the ranked code still would print in cryptographic form, e.g., "56EU" for "ДЕНЬ" A fast transliteration routine developed by Andrew Kahr for converting ranked code into a standard transliteration code has proved satisfactory for experimental purposes. It yields, for example, "DEN'" for "ДЕНЬ" .

Relatively few physical changes were necessary to achieve the desired modifications. Specially prepared keytops labelled as in Figure 2 had to be substituted for the normal ones. Corresponding type slugs were not available on the market, but were cast by the manufacturer from dies specially cut to our specifications. The correspondence between typewriter keys and the machine tokens is established physically by a set of encoding bails, notched in the pattern described in Figure 2. A photograph of the bail associated with the leftmost column of binary coding (Column 1) is shown in Figure 5. These bails were cut in our shop from blanks provided by the manufacturer, who undertook to harden the cut bails to his own specifications. Installing keytops, type slugs, and bails presented no unusual difficulties.

The author wishes to express his appreciation to the Remington Rand Univac Division of Sperry Rand Corporation, in the persons of Messrs. Edward L. Fitzgerald and Ted Carp, for their cooperation, especially in casting type slugs to our specifications, and to Messrs. Allen Christensen and Daniel Spillane of the Staff of the Computation Laboratory for machining the bails.



An Encoding Bail

Figure 5