Full Cyrillic: How Many Languages?

Olga G. Lapko
Russia, 129820, Moscow
Pervy Rizhsky per., 2
Mir Publishers
olga@mir.msk.su

A Brief History of Cyrillic

The Slavonic writing was invented by St. Cyrill and St. Method. Now there are two well-known Slavonic writings: Glagolitic and Cyrillic.

Historians are not sure whether the author of both was St. Cyrill or whether Cyrillic script was created by St. Method while St. Cyrill invented the Glagolitic alphabet. In any case, in this paper we will deal about the alphabet nowadays called "Cyrillic."

The birthday of Cyrillic is considered to be the end of May 863. May 24 was declared by UNESCO as the day of Cyrillic. By coincidence, the first conference of the Cyrillic TEX Users Group, *CyrTUG*, was held May 24–25, 1991.

The Cyrillic alphabet is based on the Greek alphabet. There were 43 letters in this alphabet. Up until the beginning of the 20th century, four additional letters existed; these are absent in the modern Russian alphabet: 'ѵ', 'ѳ', 'ѣ', and 'i'.

Nowadays Cyrillic script is used not only by Slavonic people, but also by other nations of the former USSR. Historically, many of these nations used other scripts. Some Soviet republics such as Middle Asian republics, Azerbaijan, and the Russian autonomic republics, used the Arabian script. In Siberia, the old Mongolian vertical script was used in the Buryat language, and the Dzayapandin vertical script was used in the Kalmyk language. Soon after the October Revolution many languages started to use the Latin script with additional letters. In Abkhasia, the Georgian alphabet was used for a few years. At the end of 20s and 30s, almost all languages of the USSR changed from using Latin script to Cyrillic. In many languages new letters were created (see fig. 1).

Outside Russia, Cyrillic is used in Bulgaria, Serbia, Macedonia and Mongolia. The Bulgarian language now uses only letters of the modern Russian alphabet, but earlier 'ѫ' and 'ѣ' were also used. In Macedonian and Serbian the following additional letters are used: 'ј', 'љ', 'њ', with 'ѓ', 'ќ', 'ѕ' in Macedonian only, and 'џ', 'ћ', 'ђ' in Serbian. The Mongolian language uses Russian letters with two additions: 'ө', 'ү'.

One may also find Cyrillic letters used in scripts based on the Latin alphabet. Examples are the Chinese languages: Y, Lahu, Lisu, Myao, Juang, as well as several African languages.

History of the full Cyrillic font project

The LHFONTS[1] package was created as a part of the *CyrTUG*-EmTEX package, which is distributed among Russian and non-Russian users who use Cyrillic. LHFONTS offers the `LH` Cyrillic font family; these fonts are based on the WNCYR fonts of the CYRILLIC package — part of AMS-TEX.

The main task of the LHFONTS package was to create the Cyrillic fonts family, an extension of standard text fonts of Computer Modern, which also corresponds to Russian typesetting traditions.

First this package offered two more or less popular encoding schemes:[2] Alternative — an 8-bit Latin-Russian font encoding analogous to MS-DOS's Code Page 866, mainly used by Russian MS-DOS users; and the Washington or WNCYR — a 7-bit encoding for typesetting with transliteration, which is mainly used by non-Russian users.

The Cyrillic character encodings are described in special files — `lbcoding.mf` (Alternate encoding) and `wncoding.mf` (WNCYR encoding). One can choose between these files by changing the value of one of the following variables: `altcoding`, `vfcoding` or `wncoding`. These variables also determine the font layout:

`altcoding` Standard Computer Modern in the lower part of table plus Russian letters and additional punctuation marks in the upper one; Alternate encoding: encoding file `lbcoding.mf`

[1] This package was originally named MAKEFONT, but was renamed to avoid confusion with the utility of the same name on the 4AllTEX CD-ROM, produced by the Nederlandstalige TEX Gebruikersgroep.

[2] The package also offered virtual encoding: rather conditional 7-bit encoding which combines Cyrillic and Latin fonts.

`vfcoding`	Russian letters and added punctuation marks in the upper part of table — for the following combining with Computer Modern in virtual font; Alternate encoding: encoding file `lbcoding.mf`
`wncoding`	Cyrillic letters for Slavonic languages with necessary input ligatures, standard and additional punctuation marks in the lower part of table; WNCYR encoding: encoding file `wncoding.mf`

The values of these variables are determined in driver files (`ld??font.mf`) by default. The header of these files (for ex. `ldrmfont.mf`) contains the following lines:

```
if unknown wncoding: wncoding:=0; fi
if unknown vfcoding: vfcoding:=0; fi
...
altcoding:=1-wncoding-vfcoding;

if wncoding<>0: input wncoding;
else: input lbcoding; fi
...
```

Variables `altcoding`, `vfcoding` and `wncoding` may be set by hand in the file header or at the start of the METAFONT run.

The LHFONTS package contains font headers named `lh*.mf` and `ll*.mf`

The files `lh*.mf` (56 files) are virtually identical to `cm*.mf` except for the last line:

`generate <`*driver-file*`>`

that is, the standard Computer Modern *driver file* was changed to the analogous file for the LH fonts. These file headers generate a full 8-bit Latin-Cyrillic font.

The files `ll*.mf` (also 56 files) contain only the following line:

`vfcoding=1; input <`*header-file lh**`>;`

thus the command `vfcoding:=1;` sets generation of Russian letters and punctuation marks only.

Since the WNCYR encoding was an optional encoding in this package, the files `wn*.mf` were not created, but documentation explains how to create fonts with the WNCYR encoding using a similar one-line header file as follows:

`wncoding=1; input <`*header-file lh**`>;`

The LH Cyrillic font family offers typesetting in Russian and other languages using the Russian part of the Cyrillic alphabet — that is Virtual and Alternate encoding.

The WNCYR encoding also offers typesetting in modern Slavonic texts using Cyrillic and 19th century Russian text. But there are a lot of languages which use the Cyrillic alphabet with added letters. The following sections discuss this problem.

The Global Cyrillic font as the material for Ω project

Some time ago the multilingual project Ω was started. One of the authors of Ω, Yannis Haralambous, began to create a full Cyrillic font. He offered this font to *CyrTUG* for further work.

The data for this font was taken from the Cyrillic part of the Unicode table.[3] But this table still does not cover all Cyrillic letters; some old Cyrillic letters and national letters are missing. Probably the full assortment of accented vowels is necessary, which are not included in Unicode.

The font created during this work had more than 256 letters and marks. This font assortment should be further extended and improved. During the testing of this font, I created shortened variants, or split it into a few fonts. The two Cyrillic Unicode fonts were extended with glyphs for characters not included in Unicode and created by the methods described in this paper (see the Appendix).

The methods of partial font creation may usefully improve the economy of use of the computer's memory. One may create a big 256-letter (Unicode) font and then use a virtual font to achieve the necessary encoding. Alternatively, one may create the font immediately in its required encoding.

How TEX helps METAFONT. Creation of coding and ligature-kerning tables

To create a font, METAFONT needs program descriptions for letters (a lot of them), information about lettercodes, and kerning and ligature data.

Now there are a few well-known encodings of Cyrillic. They differ in which characters they hold and in what order (see the Appendix). So, for every encoding, a separate file is needed.

Since TEX cannot use a font containing ligatures or kerning information relating to external characters, we cannot use the same table, with all Cyrillic letters, for every font: we must create a separate table for each encoding. There are five tables for the different font shapes of the text fonts of the Computer Modern family: they are included in the driver files. We must create the same number of tables for every encoding.

[3] *Unicode* — International Standard ISO/IEC 10646–1, first edition, 1993.05.01. Information Technology — Universal Multiple Octet Coded Character Set (UCS) Part 1: Architecture and Basic Multilingual Plane (Table 11, Row 04, Cyrillic).

Furthermore, for each font and each encoding we must create a header file (the Computer Modern family has 56 text fonts).

As you can see the number of files will be very large, and they will often duplicate each other.

The best solution is to create three files which contain all the information that could potentially be duplicated. The first one contains a table with all the Cyrillic glyphs and signs and all supported encodings. The second one contains data on ligature and kerning for the complete font repertoire. The third file contains the table of font names and sizes and the necessary command lines for every font header file. The data for every font in every encoding would be taken from these files and the necessary header files created. All these files are created by TEX. TEX also can create the file containing all uccodes, lccodes and mathcodes for a given font.

Preparing font headers As mentioned above, we use the parameters of the Computer Modern text fonts for creating Cyrillic fonts. First, the header files for the LHFONTS package were copied from header files of the Computer Modern family, changing the only line (See the section entitled "History of the full Cyrillic font project"). To avoid unnecessary duplication, we create header files which load the necessary Computer Modern header file, substituting the standard driver file with the driver of LH Cyrillic fonts. This task, for example, was solved in the Polish fonts package, in there is a special tricky file fik_mik.mf which substitutes standard drivers with Polish ones. One of the authors of this file, Boguslav Yackovski, allowed us to use this file in the LHFONTS package.

Now it is necessary to create header files for font creation which include one or a few lines only.

The EmTEX package supports a command operator in its MFJob program, which enables one to write necessary short commands for the METAFONT run. By using this, we can avoid creation of a lot of files with the only line:

```
input fik_mik_; use_driver;
```

For font generation on other platforms we must create these files in any case. For quick generation of the files I used the file dcstdedt.tex from DCFONTS. The original file includes the table with all font names and font sizes. We modified the file, providing a possibility to add the line (\mainfontspecific macro), which can switch on variables vfcoding:=1; or wncoding:=1; when necessary, and the parameter for a few fonts, which switch on necessary shape. To specify usage of different encodings we must change the font names, so the first two letters are changed to macro \fonttwoletters; these two letters are set by the user according to the necessary encoding at the start of TEX's run. A fragment of such a file for font headers is shown below:

```
% by default full Cyrillic Font (Unicode)
% is generated
\ifx\mainfontspecific\undefined
 \def\mainfontspecific{vfcoding:=1;}\fi
\ifx\fonttwoletters\undefined
 \edef\fonttwoletters{uc}\fi

\long\def\FontsToBeGenerated{
\tablevalues                                 %
    ( ... 8 9 10 ... 17.28[17] )             %

\makefont\fonttwoletters r                   %
    ( ... 8 9 10 ... 17.28[17] )()
    ...
\makefont\fonttwoletters tt                  %
    ( ... 8 9 10 ...          )(specific:=0;)
}
```

By default there is creation of a set of file headers and encoding for the global Cyrillic font in Unicode (see fig. 1) in this file and the file of encoding data.

Preparing the encoding file and files of ligature/kerning tables The encoding files are created from the file which contains the table of all Cyrillic glyphs and signs and all well-known (or at least necessary) encodings.

```
% by default full Cyrillic Font (Unicode)
% is generated
\ifx\fonttwoletters\undefined
 \def\fonttwoletters{uc}\fi
\def\nolettercode{*}
\long\def\CodesToBeGenerated{
\tablevalues              ( uc lh wn     ... )

\makecod CYR_A    CYRA    ( 10 80 41[A]  ... )
\makecod CYR_BE   CYRB    ( 11 81 41[B]  ... )
...
\makecod CYR_LJE  CYRLJE  ( 09 *  01[LJ] ... )
}
```

We can see that this table is analogous to the previous one. Macros, analogous to macros for creating font headers, were used for creating the encoding files.

Now we must create tables of ligatures and kerns. In the Computer Modern fonts these tables are in the font driver files. In the LH fonts the tables are in separate files. As we said above, we need to create five tables of ligatures and kernings for text fonts: 1) for roman and sans serif shape; 2) for italic shape; 3) for caps and small caps shape, for which two tables are actually necessary, separately for the uppercase and small saps letters; and 4) for large fonts like `cminch`.

The Cyrillic font is very large, but one may see that almost all Cyrillic letters can be identified in a few shape groups. We determined 14 groups for uppercase letters, 14 groups for lowercase letters and 17 groups for italic letters in Cyrillic font. The letters in these groups sometimes repeat the shape (or contour) of Latin ones, so one may use the Computer Modern table as a base.

The file of kerning and ligature data retains the shape grouping of letters, so every new letter will be added to an appropriate group. At the beginning of each group we place a “typical” Latin letter as a comment. When the new letter appears it may be added into a necessary group:

```
\writeLig{if wn:}
\writeLig{ ligtable CYR_ZE: "1"=:CYR_ZHE,
 "H"=:CYR_ZHE, "h"=:CYR_ZHE;} % "Z"
\writeLig{fi}

\Ligtab %A
\Letter{CYR_A} \Letter{CYR_A_acute}
\Letter{CYR_LIT_YUS}
...
%b
\Letter{CYR_HARD_SIGN}\Letter{CYR_YATZ}
...
%R
\WriteLig{if serifs:}
\Letter{CYR_BIG_YUS}
...
\WriteLig{fi}
%
%O
\Kern{CYR_O}{k#}\Kern{CYR_O_lcomma}{k#}
\Kern{CYR_O_acute}{k#}
...
\Kern{CYR_ABKH_O}{k#}
%
...
\EndLigtab
```

We may create a font for kern testing by taking the characteristic examples from these letter groups (see Appendix).

For creation of the necessary ligature and kerning pair tables, we use data from the encoding file which was created just before them. Now the ligatures are used in WNCYR encoding only without any changes from the original Washington State University fonts.

In addition to the tables of ligature/kerning, TeX creates a uccode/lccode/mathcode file and a file `???cod.tex`, which is used by `russianb.ldf`[4].

How METAFONT generates only necessary letters TeX has thus created files necessary for encoding and ligature and kerning pair tables. Now METAFONT must generate only the necessary glyphs required for the given encoding.

From the very beginning the `LH` font family supported different encodings. Since in different coding schemes Cyrillic letters occupy different places, in character descriptions (`beginchar` command), explicit character codes have been replaced with their symbolic names. For example, the description of the lowercase Cyrillic letter ‘a’ starts with:

```
cmchar "Lowercase Russian letter a";
beginchar(CYR_a,9.25u#,x_height#,0);
...
```

In the description of uppercase letters we added the line: `if lower_case: ... fi` for redefinition of a code in the font “Small Caps”:

```
cmchar "Uppercase Russian letter A";
beginchar(CYR_A,13u#,cap_height#,0);
if lower_case: charcode:=CYR_a; fi
...
```

In plain METAFONT the definition of the `beginchar` command has the following lines:

```
def beginchar(expr c,w_sharp,h_sharp,d_sharp) =
 begingroup
 charcode:=if known c: byte c else: 0 fi;
 ...
 enddef;
```

which means that a letter with an unrecognized code number is set to position ‘0’.

For the `LH` fonts, a letter or a sign whose code is not recognized must be skipped, so the `beginchar` command is redefined in the following way:

```
let plain_beginchar=beginchar;

def beginchar(expr c,w_sharp,h_sharp,d_sharp) =
iff known c: %
plain_beginchar(c,w_sharp,h_sharp,d_sharp);
enddef;
```

What needs to be done when it is necessary to create the Cyrillic font only, but letters and signs of standard Computer Modern have got code numbers in `beginchar` so they are always determined? The three variables which were mentioned in the section on the History of the Full Cyrillic Font Project, determined three different encodings. Now they may switch on the following:

[4] The Russian language-specific file for the Babel system.

`altcoding` Standard Computer Modern in the lower part plus Cyrillic letters and added punctuation marks in necessary encoding in the upper one

`vfcoding` Cyrillic letters and added punctuation marks in the upper, lower, or in both parts of the table — for next combining with Latin part in virtual font or for full Cyrillic font creation (for example Unicode)

`wncoding` this set was not changed — Cyrillic letters for Slavonic languages with necessary ligatures, standard and additional punctuation marks in the lower part of the table; WNCYR encoding

Now the encoding files, `lbcoding.mf` or `wncoding.mf` switched on by these variables, set necessary selection of Cyrillic letters and their encoding. In fact, the file `wncoding.mf` for WNCYR encoding was not changed. The file `lbcoding.mf` switches the necessary encoding. When we have the necessary files, we can create the font.

References

[1] К. М. Мусаев, "Алфавиты языков народов СССР", Москва, "Наука", 1965.

[2] Р. С. Гиляревский, В. С. Гривнин, Определитель языков мира по письменностям, Москва, "Издательство восточной литературы", 1960.

[3] Е. И. Убрятова, Некоторые вопросы графики и орфографии письменности языков народов СССР, пользующихся алфавитами на русской основе, Москва, 1959.

[4] Московская синодальная типография, Образцы литеръ церковныхъ, российскихъ, греческихъ, латинскихъ, грузинскихъ, еврейскихъ, немецкихъ и прочихъ, находящихся въ Московской синодальной типографіи, Москва, 1826.

[5] Образцы шрифтов, Узбекская ССР. Совет народных комиссаров, Юридическое издательство, Типография, Самарканд, 1928.

[6] Татарский язык и новые информационные технологии. Выпуск 2, Издательство Казанского университета. 1995.

[7] Языки народов СССР, 5 т., Москва, 1966–68.

[8] Andrei B. Khodulev and Irina A. Makhovaya, "On TEX experience in Mir Publishers", *Proceedings of the 7th EUROTEX Conference*, Prague, pp. 37–43, 1992.

[9] Olga G. Lapko, "MAKEFONT as part of *CyrTUG*-EmTEX package", *Proceedings of the 8th EUROTEX Conference*, Gdańsk, Poland, pp. 110–114, 1994.

[10] Fry Edmund, "Pantographia containing accurate copies of all the known alphabets in the world together with an English explanation of the regular force or power of each letter, to which are added specimens of all well-authenticated oral languages; forming a comprehensive digest of phonology", Cooper and Wilson, London, 1799.

[11] Katzner Kenneth, "The languages of the world", London, Henley:

[12] The World's major languages, ed. by Bernard Comrie, London, Sydney, 1987. Rout ledge& KeganPaul, 1977.

[13] Y. Haralambous, J. Plaice, "Typesetting in the Cyrillic alphabet with Ω — The Basic Ideas", August 24, 1994.

[14] DC-Fonts, Beschreibung der Kodebelegung: TEX 256 Zeichen — internationaler Zeichensetz, 22. März 1992.

A Appendix

Figure 1: Unicode encoding; Cyrillic part

0:		Ё	Ђ	Ѓ	Є	Ѕ	І	Ї	Ј	Љ	Њ	Ћ	Ќ		Ў	Џ
16:	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
32:	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
48:	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
64:	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
80:		ё	ђ	ѓ	є	ѕ	і	ї	ј	љ	њ	ћ	ќ		ў	џ
96:	Ѡ	ѡ	Ѣ	ѣ	Ѥ	ѥ	Ѧ	ѧ	Ѩ	ѩ	Ѫ	ѫ	Ѭ	ѭ	Ѯ	ѯ
112:	Ѱ	ѱ	Ѳ	ѳ	Ѵ	ѵ	Ѷ	ѷ	Ѹ	ѹ	Ѻ	ѻ	Ѽ	ѽ	Ѿ	ѿ
128:	Ҁ	ҁ	҂	҃	҄	҅	҆									
144:	Ґ	ґ	Ғ	ғ	Ҕ	ҕ	Җ	җ	Ҙ	ҙ	Қ	қ	Ҝ	ҝ	Ҟ	ҟ
160:	Ҡ	ҡ	Ң	ң	Ҥ	ҥ	Ҧ	ҧ	Ҩ	ҩ	Ҫ	ҫ	Ҭ	ҭ	Ү	ү
176:	Ұ	ұ	Ҳ	ҳ	Ҵ	ҵ	Ҷ	ҷ	Ҹ	ҹ	Һ	һ	Ҽ	ҽ	Ҿ	ҿ
192:	Ӏ	Ӂ	ӂ	Ӄ	ӄ			Ӈ	ӈ			Ӌ	ӌ			
208:	Ӑ	ӑ	Ӓ	ӓ	Ӕ	ӕ	Ӗ	ӗ	Ә	ә	Ӛ	ӛ	Ӝ	ӝ	Ӟ	ӟ
224:	Ӡ	ӡ	Ӣ	ӣ	Ӥ	ӥ	Ӧ	ӧ	Ө	ө	Ӫ	ӫ			Ӯ	ӯ
240:	Ӱ	ӱ	Ӳ	ӳ	Ӵ	ӵ			Ӹ	ӹ						

Since the Latin part is unchanged and uses the TEX encoding scheme the next examples show only the Cyrillic part of the font.

Figure 2: Cyrillic letters which are not included in Unicode

0:	У̀	ꜰ	Ж̧	ж̧	З̧	з̧	К̧	к̧	Н̧	н̧	С̧	с̧	Х̧	х̧		
16:	З̌	з̌	Ꙋ	ꙋ	Ҫ	ҫ	Ꙙ	ꙙ			Ꙗ	ꙗ	Л	л		
32:	Г̦	г̦	Л̦	л̦	І̦	і̦	Т̦	т̦	О̦	о̦	У̓	у̓	У̀	у̀	Э̄	э̄
48:	Ю̄	ю̄	Я̄	я̄	Ѳ̄	ѳ̄	Ү̄	ү̄								
64:	˘	ᵕ	¨	´	҃	⁀	ˢ	῎	῍	᾿	,	ʿ	΅	ˍ́	῞	˘
80:	А́	а́	Е́	е́	Е̋	е̋	Є́	є́	И́	и́	І́	і́	І̋	і̋	О́	о́
96:	У́	у́	Ы́	ы́	Ъ́	ъ́	Э́	э́	Ю́	ю́	Я́	я́	Ү́	ү́	Ә́	ә́
112:	Ѳ́	ѳ́	Ӕ́	ӕ́												
128:	Ѣ́	ѣ́	Ѵ́	ѵ́	Ѫ́	ѫ́	Ѧ́	ѧ́	Ѭ́	ѭ́	Ѩ́	ѩ́	Ѥ́	ѥ́	Ꙋ́	ꙋ́
144:	Ꙋ́	ꙋ́														
160:	«	»	№													
176:																
192:																
208:																
224:																
240:				ı												

Figure 3: MS DOS cp866 encoding

128:	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
144:	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
160:	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
176:																
192:																
208:																
224:	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
240:	Ё	ё	Є	є	Ї	ї	Ў	ў			«	»	№			

Figure 4: Washington encoding

0:	,	Љ	Џ	Э	І	Є	Ђ	Ћ	њ	љ	џ	э	і	є	ђ	ћ
16:	Ю	Ж	Й	Ё	Ѵ	Ѳ	Ѕ	Я	ю	ж	й	ё	ѵ	ѳ	ѕ	я
32:	¨	!	”	Ѣ	˘	%	´	’	(	)	*	ѣ	,	-	.	/
48:	0	1	2	3	4	5	6	7	8	9	:	;	«	ı	»	?
64:	˘	А	Б	Ц	Д	Е	Ф	Г	Х	И	Ј	К	Л	М	Н	О
80:	П	Ч	Р	С	Т	У	В	Щ	Ш	Ы	З	[	“	]	Ь	Ъ
96:	‘	а	б	ц	д	е	ф	г	х	и	ј	к	л	м	н	о
112:	п	ч	р	с	т	у	в	щ	ш	ы	з	–	—	№	ь	ъ

Figure 5: KOI-8 (Unix platform) encoding

128:					«	»	№									
144:																
160:				ё												
176:				Ё												
192:	ю	а	б	ц	д	е	ф	г	х	и	й	к	л	м	н	о
208:	п	я	р	с	т	у	ж	в	ь	ы	з	ш	э	щ	ч	ъ
224:	Ю	А	Б	Ц	Д	Е	Ф	Г	Х	И	Й	К	Л	М	Н	О
240:	П	Я	Р	С	Т	У	Ж	В	Ь	Ы	З	Ш	Э	Щ	Ч	Ъ

Figure 6: ISO 8859-5 encoding

128:	«	»														
144:																
160:		Ё	Ђ	Ѓ	Є	Ѕ	І	Ї	Ј	Љ	Њ	Ћ	Ќ		Ў	Џ
176:	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
192:	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
208:	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
224:	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
240:	№	ё	ђ	ѓ	є	ѕ	і	ї	ј	љ	њ	ћ	ќ		ў	џ

Figure 7: Apple Macintosh encoding

128:	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
144:	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
160:			Ґ					І				Ђ	ђ	ҭ	Ѓ	ѓ
176:					і		ґ	Ј	Є	є	Ї	ї	Љ	љ	Њ	њ
192:	ј	Ѕ						«	»			Ћ	ћ	Ќ	ќ	ѕ
208:									Ў	ў	Џ	џ	№	Ё	ё	
224:	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
240:	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	

Figure 8: Windows 1251 encoding

128:	Ђ	Ѓ		ѓ							Љ		Њ	Ќ	Ћ	Џ
144:	ђ										љ		њ	ќ	ћ	џ
160:		Ў	ў	Ј		Ґ			Ё		Є	«				Ї
176:			І	і	ґ				ё	№	є	»	ј	Ѕ	ѕ	ї
192:	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
208:	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
224:	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
240:	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я

Figure 9: Example of national encoding: Tatar encoding

128:	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
144:	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
160:	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
176:																
192:																
208:																
224:	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
240:	Ё	ё	Ә	ә	Ө	ө	Ұ	ұ	Җ	җ	Ң	ң	Һ	һ	«	»