L2/17-179

On the exchange of clock face information in plain text and implications for encoding

Eduardo Marin Silva

28/05/2017

I am proposing to encode a new roman numeral and three precomposed numbers in the Number Forms block

Introduction. In an age where digital clocks are not only portable but affordable one would expect that analog clocks would become obsolete. After all analog clocks require more effort to read the time and less people are being taught how to read them. However, they do have some advantages, analog clocks can be customized in very creative ways that a digital display simply cannot compete http://www.boredpanda.com/cool-and-creative-clocks/ and they are very easy to maintain because it is very common for wall mounted clocks to all share the same mechanism, so if the clock stops working, one can just buy a new mechanism and install it oneself, but digital clocks require training to repair.

Take into account that when I say analog or digital I'm referring to the display not the underlying timekeeping mechanism.

Indeed, the image of an arrangement of numerals around a circle and two hands that from the center point to them, is so associated with the concept of time itself, that even so called "smart watches" which have an entire digital screen to work with, instead offer emulated analog clock faces https://facerepo.com/app/.

The demand for representations of this very image pushed legacy standards to include symbols with different times of analog clock faces, which is now encoded in Unicode from 1F550 to 1F567. These symbols are also handy for pedagogical purposes.

The importance of plain text representation. They are four main ways to represent the numerals around the clock face:

- Abstract: The clock face is devoid of numerals and the actual time is indicated in an indirect way (markings may be present).
- Semiabstract: At least one numeral is present but not all twelve
- Numeric: The clock face utilized the normal numerals 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12
- Roman: The clock face utilizes Roman numerals usually composed of Latin letters sequences I, II, III, IV, V, VI, VII, VIII, IX, X, XI, XII (usually associated with more antique clocks)

We are not interested in abstract clock face representations, however for the rest, the symbols are composed by already existing text elements so it is only natural to want to use plain text to represent them.

As I already discussed in my proposal for an alternative roman numeral four, whatever was the rationale for including the roman numerals 1 through 12, it has had serendipitous benefits, mainly clock designers can share their clock face of roman numerals without having to change font, and it fits nicely in vertical text representations, it also allows to change the glyph to reflect the possible rotation that happens in the actual clock and finally it allows to contrast the design with any other font of regular Latin letters.

That almost solves the problem for clock designs with roman numerals, but what about input? No current keyboard can input the characters in the Number Forms block, so a good way to get those characters would be to type the regular Latin sequence and use a script to automatically replace the corresponding characters. That becomes problematic for the number four since it has two preferred representations IV and IIII. One could argue that since both representations are not used at the same time, that one just has to change the glyph of U+2163 to be like IIII, but this would be a nuisance to developers, since they would have to remember that for their font they would have to input a completely different sequence unrelated to their glyph (please see the original proposal for further rationale: http://www.unicode.org/L2/L2017/17092-roman-alt-four.pdf).

Numeric clock faces in plain text. To use a numeric clock face would in theory be easier, apart from the trouble of having to change fonts to leave other numeric sequences intact. However, we face the same problem that Roman numerals faced in vertical text, in that the numbers 10, 11 and 12, occupy two spaces instead of the expected one. Mapping it to the roman numerals greatly corrupts the meaning of those characters as well as changing the glyphs of turned number two and turned number three. Even leaving the issue with vertical text aside, it is very common for clock faces to make glyphs for those characters that are very different from their composition and the possible rotation of the characters is impossible to do. So the true solution would be to encode those three numerals as atomic characters.

Proposed characters.

Extra roman character:

U+218C IIII ROMAN NUMERAL ALTERNATE FOUR

≈ 0049 I 0049 I 0049 I 0049 I

Pre-composed numbers for clock face design:

U+218D 10 PRECOMPOSED NUMBER TEN

≈ 0031 1 0030 0

U+218E 11 PRECOMPOSED NUMBER ELEVEN

≈ 0031 1 0031 1

U+218F 12 PRECOMPOSED NUMBER TWELVE

≈ 0031 1 0032 2

Possible issues. Note that we have dropped **iiii SMALL ROMAN NUMERAL ALTERNATE FOUR**, since that number is unattested and clock faces always use the capital letters, however it is not out of the question that in the future someone may find a use for such a character, primarily to keep the sequence tables neat. It may be problematic since it is an uppercase, if it is encoded without its case pair it would be more difficult to add it later. If it is decided that both characters should be included in the same version of the standard, extra code space must be found one way or another; maybe through the creation of a Number Forms Extended block.

Also the addition of the precomposed numbers means that characters such as **U+1F51F KEYCAP TEN** would now have a decomposition as U+218D U+20E3. Something to consider. Others include **U+2469 CIRCLED NUMBER TEN**, U+246A **CIRCLED NUMBER ELEVEN**, U+246B **CIRCLED NUMBER TWELVE**

Attacking the arguments against ROMAN NUMERAL ALTERNATE FOUR. It was mentioned during discussion of my proposal that only uppercase examples were found, so I dropped the lowercase character in sacrifice of elegance. It was also noted that "*The rationale for encoding these forms is based in part on the CJK compatibility characters for Roman numerals (U+2160..U+217F), but this is not, in our opinion, strong evidence.*" But I never however used those characters as "evidence" for my proposal, the evidence was included in the form of distinct clock faces, which was the relevant part (maybe I should have been more descriptive of the relevance of the clock faces, so I have not made the same mistake here).

It was also said "*As noted in the last example on the last page, CCCC is handled as four C characters. We think a similar approach would apply here as well: users should just use four I or i characters*" However the issue at hand was never a representation of the glyphs alone, but rather its representation in vertical text and the difference it would make for clock face designers to share their designs through a font. The CCCC example does not merit encoding, but not because we are using 4 instances of U+0043 (a character without numeric value) but because we can use 4 instances of U+216D, so that way the numeric information is preserved in the backend, using only Latin letters is only acceptable if one does not require the software to recognize the quantity represented and/or simply cannot type them.

It was quoted then *"(Cf. Chapter 22 of TUS, "For most purposes, it is preferable to compose the Roman numerals from sequences of the appropriate Latin letters.")*" However, if this were true, the ARIB standard would have never required to encode those roman numerals. We have given several reasons why this proposal would be more useful than problematic, so the committee should attack those premises, not quote a document that is by no means stable in the long run.

Examples. For evidence of the alternative number four please see my previous proposal on the matter.

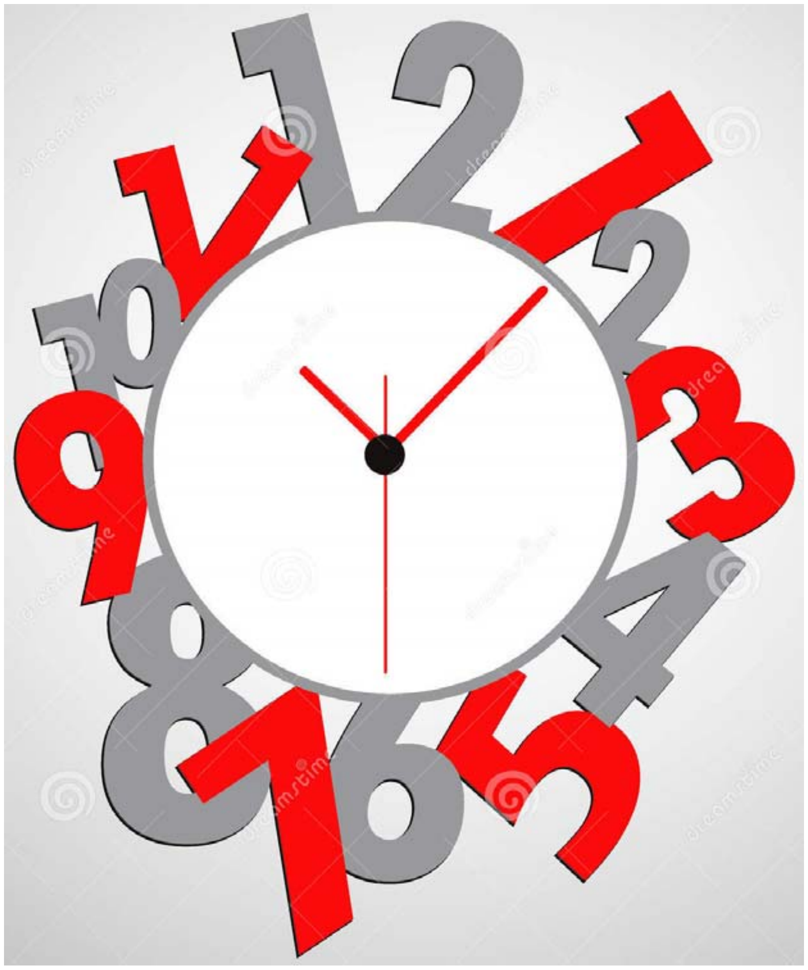

Figure 1. Example of a complex clock face. Note how we can't compose number ten eleven or twelve since they have different orientation and also different color

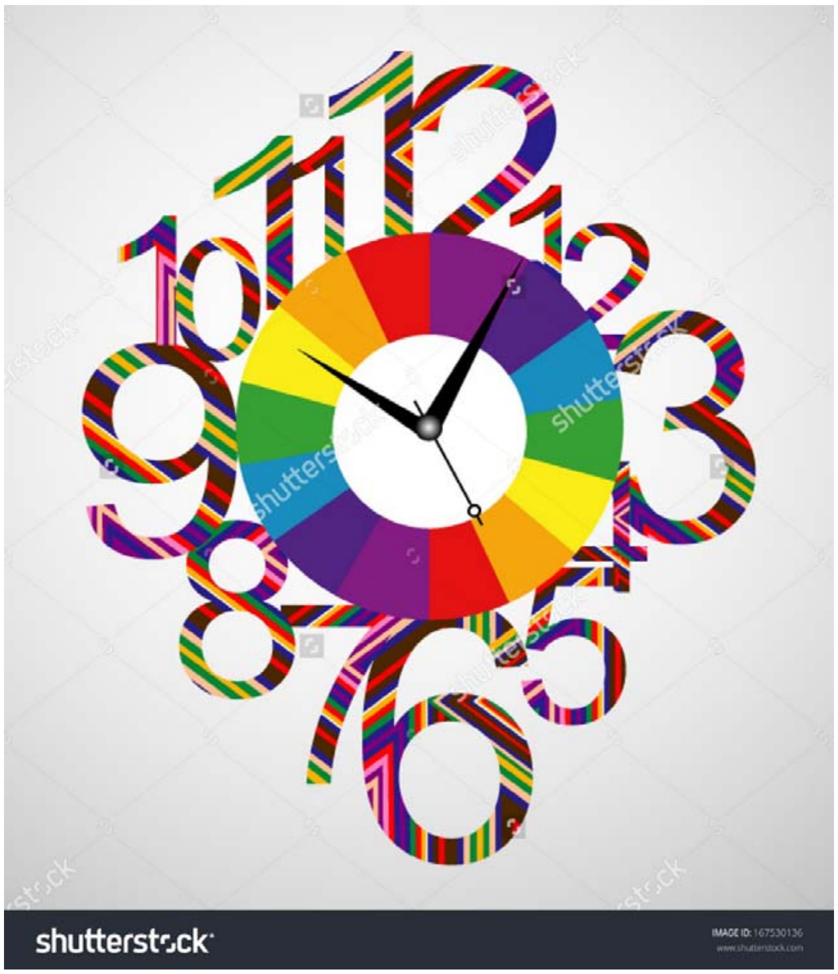

Figure 2. Another less complex face: Note that we can still cannot compose 10, 11, or 12, since the number 1 and 2 are differently sized in different contexts.

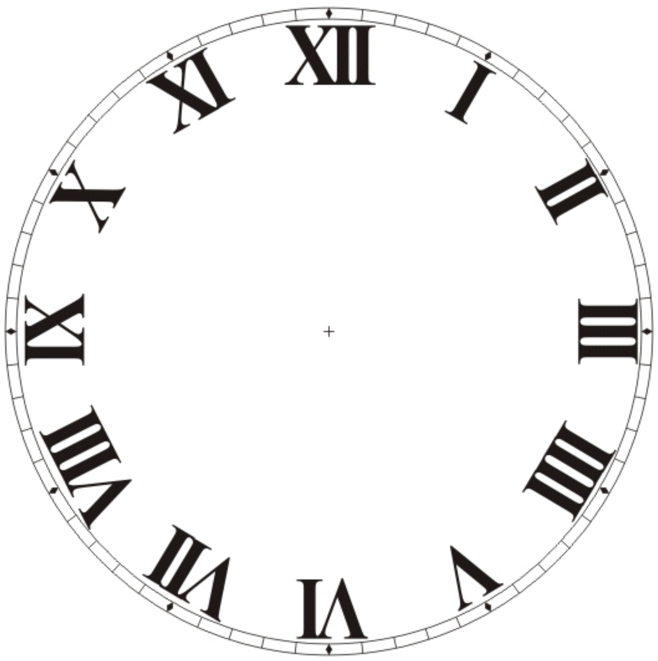

Figure 3. Clockface with numeral rotation. This rotation can easily be captured by a font by rotating the glyphs for the Roman characters

Number to be represented in vertical text	Without pre-composing	
	Roman	Numeric
1	I	1
2	I	2
	I	
3	I	3
	I	
	I	
4	I	4
	V	
4 (alt)	I	-
	I	
	I	
	I	
5	V	5
6	V	6
	I	
7	V	7
	I	
	I	
8	V	8
	I	
	I	
	I	
9	I	9
	X	
10	X	1
		0
11	X	1
	I	1
12	X	1
	I	2
	I	

Number to be represented in vertical text	With pre-composing	
	Roman	Numeric
1	I	1
2	II	2
3	III	3
4	IV	4
4 (alt)	IIII	n/a
5	V	5
6	VI	6
7	VII	7
8	VIII	8
9	IX	9
10	X	10
11	X1	11
12	XII	12

Figure 4. Notice all the vertical space saved by pre-composing the Roman numerals and numbers 10, 11 and 12.