



**HAL**  
open science

## Typesetting the Holy Qur'ān with TeX

Yannis Haralambous

► **To cite this version:**

Yannis Haralambous. Typesetting the Holy Qur'ān with TeX. 3rd International Conference and Exhibition on Multilingual computing (Arabic and Roman script), University of Durham, Dec 1992, Durham, United Kingdom. pp.110-125. hal-02100356

**HAL Id: hal-02100356**

**<https://hal.science/hal-02100356>**

Submitted on 15 Apr 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Typesetting the Holy Qur'ān with T<sub>E</sub>X

Dr. Yannis Haralambous\*

## Abstract

*Al Amal* is a typesetting system for languages using the Arabic alphabet (Arabic, Persian, Urdu, Ottoman Turkish), integrated in the T<sub>E</sub>X typesetting environment. Based on a dual approach, it is suitable both for modern as for classical fully diacriticized texts. The Naskhi typeface included in *Al Amal* has been designed with the METAFONT programming language and features more than a thousand ligatures and variant character forms. In this paper, after introductory sections to T<sub>E</sub>X & METAFONT, and a review of the traditional typographical rules of the Naskhi script as performed by *Al Amal*, we describe features concerning the typesetting of the Holy Qur'ān.

## نبذة

ان «الامل» نظام تنضيد للغات ذات الابدجية العربية مثل اللغة الفارسية، الاوردو، العثمانية واللغة العربية بالطبع. ان «الامل» متكامل ضمن نظام «تخ» للتنضيد وذو هدف مزدوج: التنضيد الحديث والتنضيد التقليدي (مع الحركات). لقد تم تصميم الحرف النسخي للطباعة في لغة «ميتافونت» ويحتوي على اكثر من الف حروف مندمجة وخاصة. في هذه المقالة نقدم، بعد نبذة مختصرة عن «تخ» و «ميتافونت»، مراجعة لاصول تنضيد حرف النسخ الطباعي في نظام «الامل» كما نصف المزايا الخاصة بطباعة القرآن الكريم.

## 1 Introduction

T<sub>E</sub>X is a programming language specialized in computer typesetting. It has been created by an eminent American computer scientist, D. E. Knuth from the University of Stanford (California) in the late seventies. T<sub>E</sub>X was first used to typeset mathematics, because mathematical formulas require two-dimensional constructions as well as structured input which cannot be achieved by an ordinary word processor based on WYSIWIG techniques.

These two features of T<sub>E</sub>X (two-dimensional constructions and structured input) are essential for the typesetting of Arabic; in fact, as *Al Amal* aims to demonstrate, T<sub>E</sub>X not only allows typesetting of Arabic, but is able lead to a revival of traditional Arabic typography by modelizing it's specific techniques.

---

\*Born in Athens (Greece) in 1962; Ph. D. on Algebraic Topology, 1990, University of Lille (France). Working areas: T<sub>E</sub>X, METAFONT, PostScript; typesetting systems and fonts for non standard languages (Greek, Arabic, Hebrew, Armenian, Slavonic, Glagolitic, Hieroglyphic Egyptian, Cambodian etc.) Member of the Technical Council of the T<sub>E</sub>X Users Group.

A traditional typographer had the possibility to use good handmade fonts; in this domain as well, D. E. Knuth created a *chef d'œuvre*, namely the programming language METAFONT. This language is specialized in the creation of fonts, combining the techniques of a traditional designer with the mathematical accuracy of the computer. METAFONT has been used to create the Naskhi font family used in *Al Amal*.

Finally, besides the artistic part of the typesetter's job, which is the displaying of text and illustrations over the page (and in the case of Arabic, the positioning of vowels and diacritics), there is also the mechanical part, which consists in selecting the appropriate types from the typecase. This corresponds to the text input, in computer lingo; *Al Amal* uses a preprocessor to convert Arabic text from any possible source into T<sub>E</sub>X code. The latter is human readable, 7-bit (and hence portable and email compatible), and can be used for typesetting as well for database operations or archiving.

The reader will find in this paper introductory sections to T<sub>E</sub>X and METAFONT, a description of the *Al Amal* typesetting system and finally a section devoted to the special typographic requirements of the Holy Qur'ān and some examples. In the bibliography he will find specialized publications and sources of inspiration for *Al Amal*, as well as a selection of introductory and more advanced literature on T<sub>E</sub>X and METAFONT (cf. the corresponding sections).

The author would like to thank Dr. Al-Dhahir (Enschede), Dr. Lagally (Stuttgart) and Dr. MacKenzie (Göttingen) for their precious help. Also he would like to thank his friend Nazih Helewa for revealing him the beauties of Arabic culture, many years ago.

## 2 What is T<sub>E</sub>X?

D. E. Knuth's T<sub>E</sub>X is the achievement of a man who believes that beauty can be found everywhere; even in the way of bringing odd mathematical formulas on paper. He not only proved that this can be done by a machine, but also that it can be done efficiently and easily. And he donated his work to the academical community by releasing it in the public domain.

T<sub>E</sub>X should be pronounced as in modern Greek τέχ; the name comes from the three first letters of words τέχνη (Art) and τεχνολογία (Technology). As every programming language it has commands and comments; as a typesetting system the main importance is given to the text. Commands are words, in most cases preceded by the character \ (backslash); everything on the right of a % (percent sign) is a comment; the rest is text.

Some characters are used for special purposes: \$ indicates mathematical text, ~ prohibits line break etc. The action of commands can be delimited to portions of text by the effect of "grouping". So for example if we want the word **liberty** to appear in bold, we will write

...we want the word {\bf liberty} to appear in bold...

where \bf is the command switching to bold, and { and } delimit the action of \bf to the word 'liberty'.

Commands can be defined or re-defined by the user to fulfill specific needs. Also portions of text can be *arguments* of commands; for example one could define a macro \arabicdate with three arguments (day, month, year), which for the values 16, June, 1992 would give ١٦ حزيران ١٩٩٢ or ١٦ يونيو ١٩٩٢. This would be written in the following way:

\arabicdate{16}{June}{1992}

where the curly braces delimit the arguments. This feature of T<sub>E</sub>X gives the user infinite possibilities of text manipulation. Mathematical symbols are also inputted by commands: to obtain

$$\int_0^{\infty} \frac{\log(x)}{|1+x|} dx$$

one can write

$$\int_0^{\infty} \frac{\log(x)}{1+x} x \, dx$$

where  $\int$  indicates mathematical display mode,  $\int$  and  $\infty$  denote the integral sign and the infinity symbol, and  $\log$ ,  $\frac{\phantom{x}}{\phantom{x}}$ , are of typographical nature: the logarithm function should appear in the text font (and not as *log*) and a tiny blank space is placed between the argument of the integral and the integration symbol  $dx$ .

A beautiful example of macro expansion in a different context, can be found in the work of T. V. Raman [1]; he converts  $\TeX$  code into *sound* by a voice synthesizer, in order to allow visually unabled persons the access to scientific documents. In his system,  $\TeX$  commands are executed (in  $\TeX$ 's lingo we say 'macros are expanded') in a way to make understanding of the formulas easier (for example  $\sum_{i=0}^n$ , written in  $\TeX$  as  $\sum_{i=0}^n$  will be naturally read 'sum for i equals 0 to n' or 'somme de i égal 0 à n' or 'ἄθροισμα γὰρ ἰ ἀπὸ 0 ἕως n', etc.)

So actually the user is *structuring* his/her text for typesetting, and this additional structure can be used in several ways. The same thing happens also at the level of the document structure. By using a very widespread set of  $\TeX$  macros called  $\LaTeX$ , the user indicates the different parts of the document, words to be included in the index, bibliographic references, cross-references etc.  $\LaTeX$  then produces a document according to 'style specifications', which can be chosen by the user or can be supplied by his/her institution to insure uniformity of design. Needless to say that the table of contents, index, bibliography, cross references, etc. are automatically generated and updated by  $\LaTeX$ .

$\TeX$  is very well adapted to multilingual texts: by using simple two-letter ISO command such as  $\backslash FR$ ,  $\backslash AR$ , anywhere inside a text the user can instantly change the hyphenation rules [this paper has been typeset using British hyphenation rules, originally implemented to  $\TeX$  out of a list of 114,925 British-hyphenated words], the punctuation spacing, the script if necessary (switching to a font of the same family, with the same characteristics), and the direction of text (the latter feature requires a special bidirectional version of  $\TeX$ , called  $\TeX-X_{3T}$ ).

Returning to the special needs of Arabic, another characteristic feature of  $\TeX$  is that it works by assembling 'boxes', in the sense that to every character of a font corresponds an invisible two-dimensional box which  $\TeX$  will very accurately place on the paper. Neither is the shape of the character known to  $\TeX$  (only the dimensions of the box), nor has the character to be inside the box: the two entities (the character and it's box) are totally independent. This fact has been used by the author in *Al Amal* for the construction of Arabic ligatures out of elementary particles which  $\TeX$  assembles with absolute accuracy by positioning the corresponding boxes; diacritics are placed by default at a height or depth determined by the dimensions of the box; in the case of the letter  $\text{g}\ddot{\text{m}}$  in fig. 1, the kasra has to be placed under the character while in the case of the  $\text{h}\grave{\text{a}}$  it can be placed inside the character: the reader can see the boxes corresponding to these letters in gray.

Last but not least a very important feature of  $\TeX$  is it's universality and portability. Written in elementary and extremely well documented Pascal, it has been ported to every possible platform, from personal computers to working stations and mainframes. D. E. Knuth has placed severe restrictions to these implementations so that they all remain strictly compatible. Hundreds of thousands of people in the world use  $\TeX$  to typeset, and to communicate their documents through email and

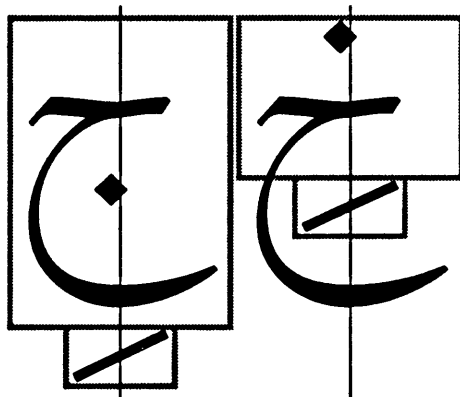


Fig 1. How  $\TeX$  assembles "boxes".

ftp; the fact of being open and public domain software has made T<sub>E</sub>X also very widespread in the Eastern and Third World countries.

For further reading on T<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X, the reader may consult [13], [15].

### 3 METAFONT

METAFONT is a programming language, as innovatory as T<sub>E</sub>X, but specialized in font design and production. In a traditional typeface, the same character can appear in different sizes, weights (light, bold etc.), and shapes (straight, slanted, small capitals etc). All of these must look close enough to show that they belong to the same font family, but again have their own specific features. Not to mention that the different characters of the same font must also have common characteristics; resemblances inside a font make it homogeneous and harmonious—differences make it lively and witty (just like human relations inside a family).

A computer should be taught about the common characteristics of characters, as well as their differences; this is exactly what METAFONT does: a ‘meta-font’ is a set of instructions by which arbitrarily many fonts can be drawn by varying the values of certain parameters. These parameters will change the size, weight and shape of the characters in different fonts, but the basic properties of the font will remain the same so that the fonts created out of a ‘meta-font’ deserve the term ‘family of fonts’.

D. E. Knuth has created a ‘meta-font’ called Computer Modern, out of which one can have fonts as various as roman, italics, typewriter, sans-serif and many more. This is perhaps — at least historically speaking — the most spectacular aspect of METAFONT, but certainly not the most important in real-world font production; after all, a font designer will barely be interested in having italics and typewriter characters out of the same code. Far more important is the possibility of adding *optical scaling* to characters: characters have thin and fat strokes; in traditional typography when these characters are scaled, the thin and fat strokes do not behave in the same way. Let’s take for example a Bodoni ‘H’; this famous Italian font designer liked to have very thin strokes (in this case, the horizontal bar) opposed to very fat ones (the two vertical stems). When this character is magnified, the thin stroke must remain thin, otherwise it would lose its elegance. On the other hand, when the character is reduced, again the thin stroke cannot be changed as much as the fat ones, because it could very easily disappear. And this is exactly what happens in nowadays DTP using scalable fonts where a single shape is reduced and magnified to produce what was specifically and separately drawn in traditional font design.

In METAFONT we are at the other extreme: every size involved in the design of a character is parametrizable, and the designer can specify the behaviour of every stroke when the font is either scaled, or when weight or shape are modified. These methods have been applied to the Naskhi font of the *Al Amal* system. In fig. 2 the reader can compare the characters ح and ل in two different sizes: 10 points and 5 points. As shown in the figure, the thin stroke width remains almost the same, while the fat strokes become thinner [the reader may be interested to know that although the characters dimensions are reduced by a factor  $\lambda = 0.5$ , the width of the fat strokes is reduced by a factor depending on the width of the thin strokes: if  $\alpha$  is the former and  $\beta$  the latter,  $\alpha$  becomes  $\lambda\alpha + (1 - \lambda)\beta$ ; this guarantees that in small sizes  $\alpha$  will tend to be equal to  $\beta$ ].

METAFONT is as portable and universal as T<sub>E</sub>X: it can produce fonts for every device (printer, fax or screen), from cheap dot-matrix printers to professional phototypesetting devices. The code written by the font designer is internally converted into Bézier curves (third degree polynomials); then METAFONT adequately rasterizes these curves according to certain parameters concerning the output device. METAFONT runs on every possible platform; special versions of it produce PostScript output as well (cf. [11], [2]).

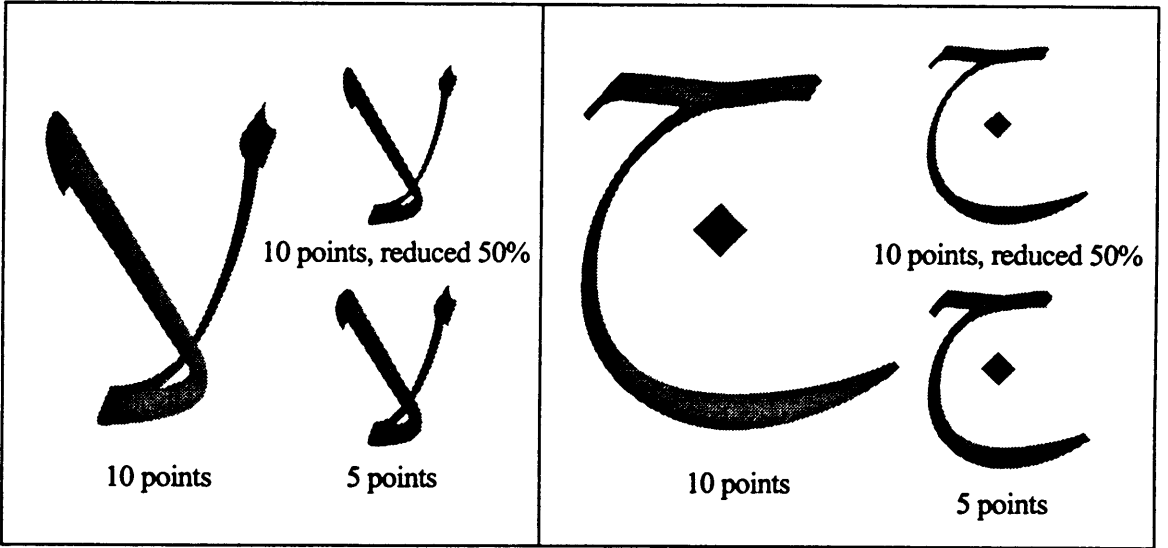


Fig 2. Linear scaling versus optical scaling

For further reading on METAFONT, the reader may consult [14], [16].

#### 4 The *Al Amal* system

There are two directions in Arabic typesetting: one is to typeset in the quickest and easiest way modern Arabic text, without vowels and diacritics (except a few essential ones, like the 'šadda', 'faḥa tanwin' etc.); the other direction is to typeset as faithfully as possible classical texts, which involve intensive vocalization and a large amount of diacritics.

These two directions are actually contradictory. Modern word processors allow an easy input of Arabic text, directly through the keyboard; but these can neither insure high typographical quality, nor can classical texts be produced: the big amount of diacritics requires several font tables of 256 characters, and a continuous font change would turn WYSIWIG input into a nightmare. Often, as we will see in the case of the Holy Qur'ān, these diacritics are quite small and must be positioned in such a delicate way that a WYSIWIG input could only be done under extreme magnification of the screen: typesetting of classical Arabic would require almost microscopical graphics work.

T<sub>E</sub>X is flexible and powerful enough to provide solutions in both directions: as long as no vowels are required, T<sub>E</sub>X can automatically typeset Arabic/Persian/Urdu text using any possible font (cf. [9]). There is no preprocessor needed, and the input is done through a simple transliteration. For example the text

يمكن تعريف التردد الجزئي (fréquence partielle) للزوج  $x_i, y_i$  وكذا التردد  
 الهامشي (fréquence marginale) للقيمة  $x_i$  (وبالتالي  $y_j$ ) بالمتساويات التالية:  
 $f_{ij} = \frac{n_{ij}}{N}$  [١.٢.١]، التردد الجزئي،  
 $f_{i\cdot} = \frac{n_{i\cdot}}{N}$  (وبالتالي  $f_{\cdot j} = \frac{n_{\cdot j}}{N}$ ) [٢.٢.١]

(from [3]) can be written through the following T<sub>E</sub>X code:

```
\AR imkn t'rif 'ltrdd 'ljz*y (\FR fréquence partielle\AR)
llzuj $x_i,y_i$ ukD' 'ltrdd 'lh'mSy (\FR fréquence marginale\AR)
llqim"h $x_i$ (ub'lt'ly $y_j$) b'lmts' uy't 'lt'li"h:
```

[1.3.1]  $f_{ij} = \frac{n_{ij}}{N}$  'ltrdd 'ljz\*y,

[1.3.2]  $f_{i\bullet} = \frac{n_{i\bullet}}{N}$   
 (ub'lt'ly  $f_{\bullet j} = \frac{n_{\bullet j}}{N}$ ).

where \AR stands for Arabic (\FA for Persian and \UR for Urdu according to ISO 359) and \FR for French. In this case, the contextual analysis as well as the ligature operations are done internally, so that they are independent and compatible with any possible macro construction (this means that Arabic text can appear inside mathematical formulas, or complicated tables, just like Latin-alphabet text). The font used in the example as well as in the abstract and bibliographic references of this paper is the standard جيزة (Geezah) font of the Apple Macintosh Arabic OS.

Before examining the other direction, namely the typesetting of classical Arabic, we will attempt to classify the difficulties involved in this task. These are of two kinds:

- those concerning the computer exclusively:
  - A direction of Arabic script;
  - B contextual analysis;
  - C text input (encoding, transliteration);
- those who were already present in traditional Arabic typography:
  - D spacing;
  - E multitude of ligatures, special symbols and variant forms of characters;
  - F positioning of vowels and diacritics.

In the case of the 'automatic' Arabic typesetting by  $\text{T}_{\text{E}}\text{X}$  described in the previous paragraph, problems A, B and C are solved entirely, D and E are solved partially and F is avoided: the direction of text is handled internally by  $\text{T}_{\text{E}}\text{X}$  (actually  $\text{T}_{\text{E}}\text{X}-\text{X}_{\text{E}}\text{T}$ ), the contextual analysis is performed automatically and the text input is done using a fixed transliteration; on the other hand, stretching of connections (كشيدة) to balance excessive blank space is done manually, some ligatures are produced automatically and there are no vowels or diacritics available, besides 'šadda' and 'fatħa tanwin'.

It is not possible to remedy to problems D – F in an automatic way, at least as long as 256-character font tables will be used (in a  $\text{UNICODE}$ -like environment the situation would be entirely different). Because of this fact, the author has chosen an entirely different approach for the main *Al Amal* system, in the case of classical Arabic. The use of more than one font tables — which are necessary for producing all characters, ligatures, special symbols and diacritics — requires a text preprocessor, which will perform the contextual analysis as well. The correct positioning of vowels and diacritics is 'taught' to  $\text{T}_{\text{E}}\text{X}$  by a process of successive approximations, similar to the way a (human) typesetter of Arabic get's trained to perform this esthetically very involving task. In sections 4.1 – 4.2.1 the main points of the *Al Amal* system for classical Arabic are described. Finally, in section 5 the aspects of *Al Amal* involving the Holy Qur'ān are more closely examined.

#### 4.1 Typographical enhancements: ligatures, variant forms

Typography in the Naskhi script has a long tradition, inspired by Islamic calligraphy. It is an amazing fact that despite the total freedom of Islamic calligraphy, typography has kept very strict and efficient rules for text enhancement. Some of these rules are mentioned in the classical paper [19] by H. Ritter; a much more detailed description can be found in [4] (the user may compare these rules to the rules of calligraphy of Hāšim Muḥammad al-Ḥaṭṭāṭ in [10]).

We will call “typographical enhancement” one of the following situations: (a) two or more (up to four) consecutive types are replaced by a single type, (b) under certain circumstances a character is replaced by a variant form; the first case will be called a *ligature*, while in the second case we will speak of *variant forms*.

To describe the various ligatures and variant forms of the Naskhi script, we will classify some letters of the Arabic alphabet into families, according to their shapes. This classification is **purely graphical**; the common characteristic of characters in the same family will be that they “react in the same way”, with respect to typographical enhancement operations (ligatures, variant forms).

These families are the following:

“alif family”: consisting of letters ا, آ, أ and إ;

“bā’ family”: consisting of letters ب, ت, ث, پ (Persian, Urdu), ٹ (Urdu), in all forms; ن, ی and ئ belong to the family only in their initial and medial form;

“ḡīm family”: letters چ, خ, ح, ج (Persian, Urdu), in all forms;

“rā’ family”: letters ر, ژ, ژر (Persian, Urdu), ر (Urdu) in both forms;

“sīn family”: letters س and ش in all forms;

“ṣād family”: letters ص and ض in all forms;

“ayn family”: letters ع and غ in all forms;

“fā’ family”: letters ف, ق, قف in initial and medial forms;

“kāf family”: letters ك, گ (Persian, Urdu), ك (Ottoman Turkish);

[the letter ك is used to distinguish the special use of ك as ‘n’ in Turkish words like *سونك son*, *يكنك*

*binlik*, *گنیشك gensištyk* (cf. [5])]

“nūn family”: the final and isolated forms of letters ن and ن (Urdu);

“hā’ family”: letters ه, ه, ه (Persian) in final and isolated forms;

“wāw family”: letters و and و;



“yā’ family”: letters **ي**, **ي** and **ئ** in final and isolated forms only;

This classification implies that an Arabic typesetting system featuring a certain ligature involving for example the letter *bā’* will have to provide the same ligature for all letters of the *bā’* family. In the case of multiple ligatures, this can lead to a big amount of possible combinations. Follows a list of ligature operations which is not exhaustive, but nevertheless covers the most fundamental cases; examples are given.

The notation  $\langle b\bar{a}' \rangle \langle \tilde{g}im \rangle mim$  means: any character from the *bā’* family (*bā’*, *tā’*, *ṭā’*, etc.) followed by any character from the *ḡim* family (*ḡim*, *ḥā’*, *ḥa’*, etc.) followed by *mīm*.

#### 4.1.1 Ligatures

##### 1. $\boxed{l\bar{a}m-l\bar{a}m-h\bar{a}'}$

The name of God requires a special and essential ligature which usually includes already the *ṣadda* and the vertical *fatha*: **لله**; in unvocalized texts the *fatha* can be omitted, in which case the *ṣadda* can be a little higher: **لله**.

##### 2. $\boxed{\langle k\bar{a}f \rangle \langle 'alif \rangle, \langle k\bar{a}f \rangle l\bar{a}m, \langle k\bar{a}f \rangle l\bar{a}m \langle 'alif \rangle}$

A character from the *kāf* family in initial or medial form, followed by a character from the *'alif* family, or *lām*, or *lām* plus a member of the *'alif* family. Examples: **كل كلمة**, **كأب**, **گل بی** (“pear” in Persian).

##### 3. $\boxed{\langle k\bar{a}f \rangle \langle b\bar{a}' \rangle \langle 'alif \rangle, \langle k\bar{a}f \rangle \langle b\bar{a}' \rangle l\bar{a}m, \langle k\bar{a}f \rangle \langle b\bar{a}' \rangle l\bar{a}m \langle 'alif \rangle}$

The analogue of the previous ligature operation, with a character from the *bā’* family between the first two letters. Examples: **بيكباش**, **كباية**, **كيلا** (“major” in Turkish).

##### 4. $\boxed{\langle k\bar{a}f \rangle m\bar{im} \langle 'alif \rangle, \langle k\bar{a}f \rangle m\bar{im}-l\bar{a}m, \langle k\bar{a}f \rangle m\bar{im}-l\bar{a}m \langle 'alif \rangle}$

Like the previous case, but with a *mīm* between the first two letters. Examples: **كان**, **كل**.

##### 5. $\boxed{\langle b\bar{a}' \rangle \langle \tilde{g}im \rangle, \langle \tilde{g}im \rangle \langle \tilde{g}im \rangle, \langle s\bar{in} \rangle \langle \tilde{g}im \rangle, \langle s\bar{a}d \rangle \langle \tilde{g}im \rangle, \langle 'ayn \rangle \langle \tilde{g}im \rangle, \langle f\bar{a}' \rangle \langle \tilde{g}im \rangle, l\bar{a}m \langle \tilde{g}im \rangle, l\bar{a}m-m\bar{im} \langle \tilde{g}im \rangle}$

These are perhaps the most common ligatures: a character in *initial form* followed by a character from the *ḡim* family, in medial or final form. In the case of the *bā’* family, the first

character has a special shape: **بچه**, **تخت**, **بجانه** (“child” in Persian), **میخ** (“ice” in Persian). In the case of the *mīm*, the latter is replaced by a smaller form: **مجد**. All

other letters remain unchanged: **صنج, صحنى, شخ, سجع, نجلان, ججج, جحفل**.

**لجام, لج, لجر, عجل, عجل**. In the case of the fā' family, the first letter must be moved to the right if the second letter is a ha': **فجر, فحل**, but **فخر**. Finally lām-mīm in front of a member of the ġīm family forms a three-fold ligature: **لمجة**.

6. **<bā'><bā'><ġīm>**

Similar to the previous ligatures, but with two members of the bā' family in front of a character from the ġīm family. Examples: **سج, سنج, سجر** ("screw" in Persian).

7. **<bā'><ġīm>mīm, <ġīm><ġīm>mīm, <sin><ġīm>mīm, <ṣād><ġīm>mīm, <'ayn><ġīm>mīm, <fā'><ġīm>mīm, lām<ġīm>mīm, lām-mīm<ġīm>mīm, <bā'><bā'><ġīm>**

The same as in the preceding two cases, but followed by a mīm. This ligature appears only in initial form (*i.e.* must be followed by a connected letter). Examples: **مجة, مجى, تحمل**.

**محمل, لمجة**.

8. **<bā'>mīm, mīm<'alif>, <bā'>mīm<'alif>**

The combination of a member of the bā' family followed by a mīm produces always a ligature; this ligature can be of initial form: **تمتع, ثمل**; of medial form: **منه, منة**; of final form: **مقيم, مقيم**; or of isolated form: **خاتم, قائم**.

The reader has encountered the mīm<'alif> ligature already as part of the <kāf>mīm<'alif> one. When preceded by a member of the bā' family, a ligature is produced in final form only: **ليان, اعنا**.

9. **<lām>mīm, <kāf>mīm**

The former combination always produces ligatures, while the latter produces ligatures only inside, or at the end of a word (medial, final and isolated positions). Examples: **لماح**,

**كم, لكم, لكمة, لم, علم, سلمى**, while **كمن** remains unligatured.

10. <gīm>mīm, <sīn>mīm, <‘ayn>mīm, <fā’>mīm, <kāf>mīm, lām-mīm, mīm-mīm, hā’-mīm

Also a very frequent group of ligatures, appearing only in initial and isolated form. The initial forms of these ligatures can be divided in two groups: those with a filled mīm, and those with a hollow one. Examples of the first case are: ممنوع, شمس, سمل. All remaining

letters belong to the second group: همج, قمع, غمط, عملاق, حمن.

The isolated forms are quite similar: هم, مم, فم, سم, جم etc.

11. <bā’>hā’, lām-hā’

The lām-hā’ ligature appears only in initial form: لهاب لهج.

Combinations of members of the bā’ family followed by ha, produce ligatures in initial and medial form (because as we know, the final form of hā’ is quite unexpectedly different).

Examples: تهرل (“wear” in Persian), پهلوی بهم.

12. <bā’><rā’>, <bā’><nūn>

These are final ligatures: the letter from the bā’ family is raised and the connection with the following letter is done higher than the baseline. Examples: چیز, فتر, ثبر (“thing” in Persian); پختن, فن, فلسطین (“cook” in Persian).

13. <bā’><yā’>, <sīn><yā’>, <ṣād><yā’>, <fā’><yā’>, <kāf><yā’>, lām<yā’>

These are final and isolated ligatures. They can be subdivided into three groups, according to the shape of the yā’

- In the first group the yā’ is long: the sīn and ṣād families belong to it. Examples:

مسی, مضیء

- In the third group the yā’ is deep and round: fā’, kāf families and the letter lām. Examples:

لی, زندگی, فلکی, فی (“life” in Persian), the final ligatures of the fā’ family,

are lowered so that the lower part of the first character loop reaches the baseline: شفی,

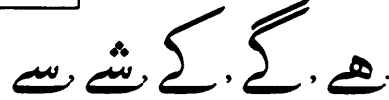
لقی.

- The second group is formed by the bā’ family: final forms are similar to those of the

first group, and isolated forms to those of the second. Examples: سبکی, سبی.

## فدائی قربی

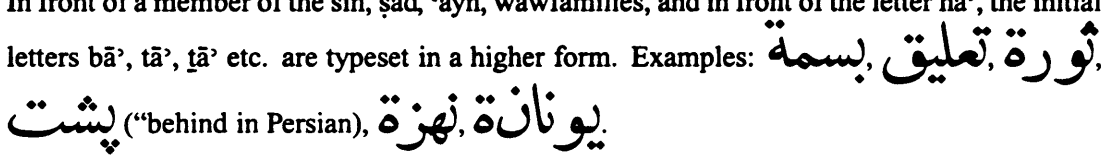
### 14. <sīn>Urdu 'alif, <kāf>Urdu 'alif, hā'-Urdu 'alif

There are five ligatures involving the Urdu variant 'alif: 

#### 4.1.2 Variant forms


The second kind of typographical enhancement found in the Naskhi tradition are variant character forms. In the following rules we will use the same notation as in the previous section; for example, <bā'><sīn> will mean: any character from the bā' family (bā', tā', t̄ā', etc.) followed by any character from the sīn family (sīn, šīn). Furthermore the variant form is underlined: <bā'><sīn> means that the letter from the bā' family will be replaced by a variant form.

#### 1. <bā'><sīn>, <bā'><ṣād>, <bā'><'ayn>, <bā'><wāw>, <bā'>hā'

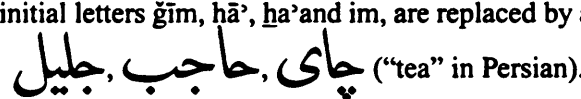
In front of a member of the sīn, ṣād, 'ayn, wāw families, and in front of the letter hā', the initial letters bā', tā', t̄ā' etc. are typeset in a higher form. Examples:  ("behind in Persian),

#### 2. <bā'><bā'><bā'>, <bā'><sīn>

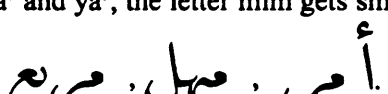
The previous rule was dealing with initial letters from the bā' family. This time the medial form is enhanced: everytime there are three consecutive members of the bā' family, the middle one should be typeset in a higher form; in front of a sīn or a šīn, the higher form is used as well [probably to distinguish between a sīn/šīn and three consecutive bā' family letters]. Examples:

 ("cotton" in Persian),

#### 3. <ḡīm><'alif>, <ḡīm>lām

In front of the 'alif or the lām, the initial letters ḡīm, hā', ha' and im, are replaced by a rounder and more closed form. Examples:  ("tea" in Persian).

#### 4. mīm<rā'>, mīmhā', mīm<yā'>

In front of a letter from the rā' family (which as the reader will see in the next rule, will be typeset in a calligraphic form), or in front of hā' and yā', the letter mīm gets smaller (like in the case of the mīm<ḡīm> ligature). Examples: 

#### 5. <ḡīm><rā'>, ṭā'<rā'>, ṣā'<rā'>, <'ayn><rā'>, <fā'><rā'>, <kāf><rā'>, hā'<rā'>, mīm<rā'>

After a letter from these families the rā', zā', že, Urdu ray letters are typeset in a calligraphic

form. In the case of the mīm, the latter is typeset in its smaller form (see above). Examples:

گربه، کریم، قرمہ، غزل، عربی، ظرف، طرطر، حرفی  
 (“cat” in Persian), هرع.

#### 6. kešide and wide characters

Since hyphenation is not allowed in Arabic script, justification on both sides can lead to excessive blank spaces between words. To remedy to this situation, connections between letters are stretched; this stretching is called کشیده (kešide). There seem to be no strict rules on this stretching operation. Following [4] (p. 102, the term for ‘stretch’ in German being „Verlängerung“), we can assert that:

- (a) stretching is *usually* done before the last letter;
- (b) one should never stretch after a single letter;
- (c) **س** and **ش** can be followed by a very wide stretch.

According to these rules, **مختلف**, **مشقف**, are correct, while **مشقف** is obviously wrong.

Another way of improving blank space balance is by using special wide characters. The most common are kāf in initial and medial form, and nūn in final and isolated form. Examples:

مکن، لبنان، وطن، مکتب، کبیر

## 4.2 Text input

After this multitude of ligatures and variant forms, the reader may be relieved to know that *Al Amal* automatically performs all these ligature and variant form operations without his/her intervention. Actually the process is splitted into two parts:

1. the preprocessor analyses the input, separates diacritics from letters, performs the contextual analysis, finds out which ligatures are available in the font (not all fonts have so many ligatures. . . ), checks out the rules of variant forms, finds out where to place potential kešides, and writes the whole information using a very strict, but human-readable syntax, where each letter, vowel, diacritic or special symbol is given by a T<sub>E</sub>X macro;
2. T<sub>E</sub>X does all the rest: it expands the macros, checks if there is a preloaded diacritic adjustment, finds out the positions of characters in the font table, and typesets them.

The first part of the process is independent of font, platform and output device. The preprocessor replaces each Arabic letter by a macro with (at least) two arguments: it’s contextual form and eventually information on the required diacritics.

From the user’s point of view, the whole process can be completely transparent; if his/her platform is able to run under batch mode, it will automatically execute the preprocessor before T<sub>E</sub>X; in this way he/she will never have to see the preprocessor’s output. The input can be in any form; for example, it can be a 7-bit left-to-right ASCII text, using an arbitrary transcription (the data on the transcription

is then written in a special configuration file), or it can be a right-to-left Arabic document created by any customary text editor, or it can be ASAMOS-encoded etc.

Let's take a closer look now to the output syntax; for the following sentence (taken from [17]):

فَغَارَتْ عَمْتُهُمْ مِنْ شِدَّةِ مَحَبَّةِ أَخِيهَا لِأَوْلَادِهِ

the preprocessor's output will be:

```
%@NASKHI
\fa1{\fatha}\ghayn2{\fatha}\alif3{\rcorr\ra0{\fatha}\ta0{\sukun}
\llig1{\ayn}{\fatha}{\mim}{\shaddafatha}\llig2{\ta}{\damma}{\ha}{\damma}\kesh\mim3{\sukun}
\mim1{\kasra}\nun3{\sukun}
\shin1{\kasra}\dal3{\shaddafatha}\tamarbuta0{\kasra}
\llig1{\mim}{\fatha}{\hha}{\fatha}\ba2{\shaddafatha}\kesh\tamarbuta3{\kasra}
\alif0{\hamzafatha}\kha1{\kasra}\llig2{\ya}{\ha}{\fatha}\kesh\alif3{
\lamalif0{\kasra}{\hamzafatha}\wcorr\waw0{\sukun}\lamalif0}{\fatha}\dal0{\kasra}\ha0{\h
```

The flag `@NASKHI` has to be present in the preprocessor input; other flags, like `@HEBREW`, `@HINDI`, `@UIGUR`, `@LATIN` switch (or will switch in the future) to the corresponding scripts. These flags still remain in the output as `TeX` comments (`%@NASKHI` etc.), so that the preprocessor can efficiently convert the code back efficiently to transliterated input.

There is one line of code for each word. In our example, the first word (فغارت) contains no ligature. The command `\fa1{\fatha}` indicates a fā', in form 1 (0 for isolated, 1 for initial, 2 for medial, 3 for final), with a fatha vowel: 1 and `\fatha` are the two arguments of the `\fa` command. Similarly, `\ghayn2{\fatha}` produces a ġayn in medial form, with a fatha vowel. Next comes `\alif3{}`. This letter is not vocalized in the text, so the second argument is empty (nevertheless the curly braces must be present, otherwise `TeX` will consider the next command as an argument to `\alif`, which will produce an error). The command `\rcorr` brings the ر closer to the ت.

On the second line we see the command `\llig` with arguments 1, `\ayn`, `\fatha`, `\mim` and finally `\shaddafatha`. As the reader has surely realized by comparing the code with the example, this command will produce an 'ayn-mīm ligature, in initial form, with a fatha on the 'ayn and a šadda-fatha on the mīm. The number of 1 in the name `\llig` gives the number of letters in the ligature (there are `\lllig` and `\llllig` commands as well).

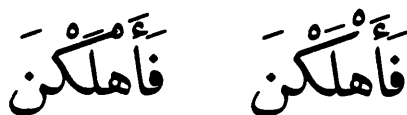
The ligature lām-'alif is a special case; `\lamalif` has three arguments: the contextual form and information on diacritics on the two letters. The command `\wcorr` brings the و closer to the ل. `\kesh` stands for the automatic stretching of letter connections (كشيدة). `TeX` will find out the exact width of connecting stroke to insert, by performing an optimization algorithm over the entire paragraph. The rest of the code is straightforward.

In addition to the features which appear in the code sample, the user has the possibility of indicating that a portion of code between the `@NASKHI` and `@LATIN` flags should not be affected by the preprocessor, by including it between angle brackets `< >`. Preprocessor directives `@IF ...`, `@ELSE`, `@ENDIF` are available as well.

#### 4.2.1 Positioning of diacritics

Diacritics are placed at a uniform height (resp. depth) over (resp. under) the character, except if the character's dimensions exceed a certain limit. The choice to place the diacritics on the same level is a personal one: the author estimates that the text is more legible, and that accents on different levels are more appropriate to calligraphy.

Sometimes adjustment of the diacritic is needed. For example in the word **فَاهِلِكُنْ** on the left, the sukūn over hā' is placed at normal height; on the example on the right it has been raised by .3 ex, where ex is the height of a lowercase 'x' (at the same magnification as the font). The fatha over the lām has been raised as well, but only by .15 ex.



These adjustments, which are necessary in special situations like this one (the cause of the adjustment is the letter kāf which is not even adjacent to hā'; this, even T<sub>E</sub>X can hardly predict. . . ), can be done “on-the-fly” by inserting adjustment commands in the code: in the case of the sukūn over the hā' this would be `\ha1{\vadj{.33ex}\sukun}` where `vadj` stands for ‘vertical adjustment’.

Other adjustments are repeated all over the text. These can be preloaded in the definitions of T<sub>E</sub>X commands, and in this way they will be executed every time a certain letter-diacritic combination occurs (for example in the case of the initial kāf T<sub>E</sub>X will automatically move an upper diacritic slightly to the left, while a lower diacritic will remain unchanged). If adjustments for a certain letter-diacritic combination have already been preloaded then eventual subsequent “on-the-fly” adjustments will be numerically added to them.




By this method, T<sub>E</sub>X can be ‘trained’ to position diacritics correctly: if the user encounters an adjustment very often, he/she will at a certain point include it to the pre-defined adjustments. This method of successive approximations is analogous to the way a traditional typesetter accumulates experience: first he/she will place the diacritics on standard default positions; next he/she will find exceptions and eventually consult a more experienced colleague; and finally he/she will develop an intuition, based on repeated operations, and will automatically apply certain adjustments and invent his/her own rules. This is what T<sub>E</sub>X does: if you don't specify any adjustment it will place the diacritics on default positions (as in the sentence above); you can specify exceptional adjustments; and you can preload adjustments and make them permanent.


## 5 Typesetting the Holy Qur'ān

The Holy Qur'ān requires a very rigorous discipline in recitation; to assist the reciter the text has been enhanced by several indications on vowels, joints, text structure, pausing, etc. A typesetting system which aims to typeset the Holy Qur'ān should provide all of these. In this section we attempt a classification from the typographical point of view, strongly inspired by the magnificent book *Introduction to Qur'ānic Script* by Syed Barakat Ahmad ([20]).

### 5.1 Diacritics


Besides the standard vowels and diacritics of modern Arabic, the typesetting of the Holy Qur'ān requires several other variant forms. Follows an attempted list, which includes the standard forms for the sake of completeness.


1. the standard fatha , kasra , damma 

2. the sukūn in two different forms: the standard one: , and the “unclosed” (cf. [20], p.17)

one: . Examples:

وَلَيْكَ وَأَعْلَيْهِمْ

3. the weṣla , which can appear only on the 'alif and eventually on the lām-'alif.



4. the madda  which unlike modern Arabic can appear on several other letters than 'alif; between them the so called muqatta'āt (isolated) letters: طس يس حم ص ق ن

طس يس حم ص ق ن  
 كهيص عسق

in the following examples madda is used together with


šadda: المص المر طسم الم

5. Variation (Middle East texts): the vertical faṭḥa , (also called 'dagger 'alif'), the kasra

followed by the small letter  on the baseline, and the ḍamma followed by the small letter  ,

under the baseline. Examples:

وَاللَّهُ عِنْدَهُ رَحْمَنٌ وَإِنْ يَشْرِكْ بِهِ، الْيَتِيمِ وَالْمَسْكِينِ

All three signs can carry a madda: , , . Examples:

اللَّهُ وَعِدَّةٌ وَإِذْ لَجْنِبُهُ يَا أَيُّهَا

6. Variation (Indo-Pakistani texts): the same vertical faṭḥa , but instead of small letters, the

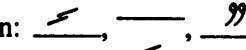
vertical kaṣra  and the inverted ḍamma . Examples: قومه وقيله

يستحي له

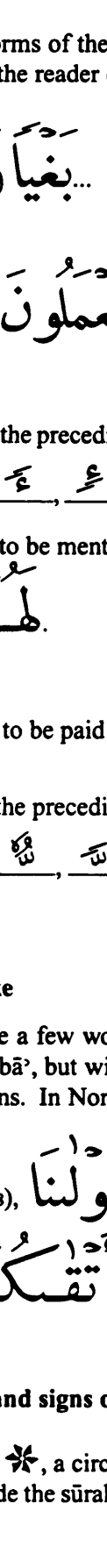
7. Nunation: faṭḥa, kasra and ḍamma tanwīn , , . The ḍamma tanwīn has a

variant form: .



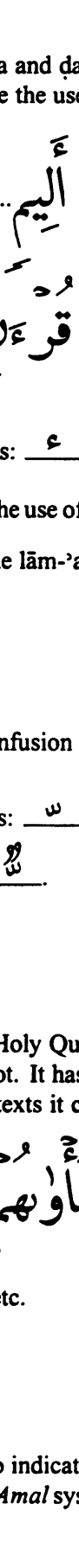
8. Nunation: variant forms of the fatha, kasra and ḍamma tanwīn: . In the following examples the reader can compare the use of standard and variant nunations:

...بَغِيَا وَعَدُوا حَتَّىٰ...  
 عَلَيْكُمْ عَذَابٌ يَوْمَ الْيَوْمِ...  
 ...مِنَ قُرْءَانٍ وَلَا تَعْمَلُونَ مِنْ عَمَلٍ إِلَّا...  
 ...

9. Hamza, with most of the preceding diacritics: . The use of hamza is straightforward; the only special case which deserves to be mentioned is the lām-'alif with 'central hamza' لا, like in

لَهُمْ فِي الْآخِرَةِ

Special attention has to be paid to avoid confusion of لا with لا.


10. Šadda, with most of the preceding diacritics:  (like in اللهُ),


## 5.2 The redundant stroke


According to [20], there are a few words in the Holy Qur'ān, which have the *redundant stroke*: a letter with the shape of the bā<sup>2</sup>, but without the dot. It has no phonetic value and has been retained only for grammatical reasons. In North African texts it carries the dots of the yā<sup>2</sup>. Here are some


examples: لَنْزَابِكْ (IV.114), نَجْوَاهُمْ (III.151), مَاوَاهُمْ (II.286), مَوْلَانَا (II.98), مَيْكَلْ (VII.60), أَتَقْسِمُ (XLIX.13), مَثْوَاهُ (XII.21) etc.

## 5.3 Punctuation marks and signs of pause










Besides the Naskhi asterisk , a circle is used to indicate the end of an āyah. This circle includes the number of the āyah, inside the sūrah. In the *Al Amal* system, a TeX macro `\ayah{}` produces this

numbered circle; the argument of the macro is the number of the āyah: for example `\ayah{89}` will produce . There are fourteen places (cf. [20], p.108) in the Holy Qur'ān where a prostration is due; in some editions the latter is indicated by the word السجدة (prostration) on the āyah circle. In

*Al Amal*, a special macro `\ayahh{\sajdah}` produces this symbol: . Another important

symbol is the  (صلى الله عليه وسلم) which is found in qur'ānic comments. Since it is used in the text (and not as an figure or illustration), it has to be in the font. In *Al Amal* it is produced by the macro `\saliallah`.

Quoting Syed, *the qur'ānic punctuation is not only based on the structure or the syntax of the sentence, but also on the need to pause, for breath and for effect*. Several signs indicate this background structure of the qur'ānic text. From the typesetter's point of view, they are signs placed on a higher level than vowels and diacritics; eventually there can be a second pause sign on the first. This requires all pause signs to be available at two levels; T<sub>E</sub>X does this with macros: for example the macro for the لازم sign will be `\lazim0` for the first level, and `\lazim1` for the second level. Since these signs don't appear very often, the user could include them by their T<sub>E</sub>X macro names in the preprocessor input, enclosed in angle brackets (cf. 4.2). Follows a list of these signs, with a short explanation taken mostly from [20]) and the T<sub>E</sub>X macro name used in the *Al Amal* system.

1. The لازم sign . *If this stop is disregarded, the meaning of the āyah may be affected.* T<sub>E</sub>X macro name: `\lazim`.
2. The مطلق sign . *Full stop, even though the āyah is not complete.* T<sub>E</sub>X macro name: `\mutlaq`.
3. The لا sign . *Forbidden pause.* T<sub>E</sub>X macro name: `\mamnoua`.
4. The سكتة sign . *Pause at this sign for a moment, hold your breath and then continue in the same breath.* T<sub>E</sub>X macro name: `\sekta`.
5. The وقفة sign . *Pause without breaking breath for a longer period than سكتة and then continue.* T<sub>E</sub>X macro name: `\waqfah`.
6. The جائز sign . *It is better to stop, though the reciter is permitted to proceed without stopping.* T<sub>E</sub>X macro name: `\jaiz`.
7. The جائز وقف sign . *A possible stop.* T<sub>E</sub>X macro name: `\jaizwaqof`.
8. The مجوز sign . *It is better not to stop.* T<sub>E</sub>X macro name: `\mujaz`.
9. The مرخص sign . *It is better to continue without stopping, but if the reciter is tired or there is any other valid reason, he may stop.* T<sub>E</sub>X macro name: `\murakhaz`.

10. The **وقف عليه** sign **ق**. *It is better not to stop.* T<sub>E</sub>X macro name: `\lawaqof`.
11. The **يوقف عليه** sign **قف**. *It is better to stop here.* T<sub>E</sub>X macro name: `\waqof`.
12. The **قديوصل** sign **صل**. *The majority view is that one should pause here.* T<sub>E</sub>X macro name: `\wasl`.
13. The **أولى أوصل** sign **صل**. *It is better not to pause.* T<sub>E</sub>X macro name: `\jaizwasl`.
14. The **نون مخفية** sign **ن**. *The 'hidden' letter nūn.* T<sub>E</sub>X macro name: `\nunmakhfi`.
15. The **معانقة** sign **❖❖**. *This sign is used in pairs to mark a word or expression which can be contrued to read either with the words preceding it or with those following it. If a pause is made on either of the two, no pause should be made on the other.* T<sub>E</sub>X macro name: `\muanaqah`.

Here is an example of the use of **معانقة**, **أولى** and **أوصل**, **جائز**:

ذٰلِكَ الْكِتٰبُ لَا رَيْبَ فِيْهِ هُدًى لِّلْمُتَّقِيْنَ

The Holy Qur'ān is arranged in 114 sūrahs (chapters); each sūrah is divided in rukū's (sections), and in āyahs (paragraphs). There is also another division of the Holy Qur'ān, namely in 30 equal parts, called juz's. The end of a rukū' is indicated by the sign **ع** (T<sub>E</sub>X macro `\ruku`). There is also a marginal sign for the same purpose, which displays also the serial number of the rukū' as well as

the number of āyahs in the rukū' and the number of rukū' in the juz'. For example, the sign **ع**

means that this is the first section of the sūrah; that the section contains ten āyahs; and that this is the fourth rukū' in the juz'. In *Al Amal* this is produced by a simple T<sub>E</sub>X macro `\bigruku{1}{10}{4}` (the example was produced by `\bigruku{1}{10}{4}`).

## 6 Conclusion

Each edition of the Holy Qur'ān has its particularities; the author has tried to cover the needs of a number of editions, from different origins. The force of *Al Amal* is that, being an open system, it can be adapted to any particular needs. New signs can be designed by the font creation programming language METAFONT; new page styles can be implemented to T<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X; and once written in form of macros by the preprocessor, the text can serve for other purposes than typesetting.

The *Al Amal* system is part of version 1 of ScholarT<sub>E</sub>X: a package of T<sub>E</sub>X macros, fonts, preprocessors and related utilities for scholars. It includes also T<sub>E</sub>X systems for Greek, Epigraphical Greek and Latin, Hebrew, Syriac, Armenian, Saxon, and Old German. Forthcoming versions will

feature Khmer, Sanskrit, Tifinagh, Uigur, Hieroglyphic Egyptian, Slavonic, Glagolitic, Irish and other scripts. ScholarT<sub>E</sub>X also serves as a platform for distribution of valuable public domain T<sub>E</sub>X software, of scholarly interest (IPA fonts, macros for scholar editions, and others).

In the appendix of this paper the reader will find a page from the Holy Qur'ān (starting at rv.78, the punctuation following [18]), entirely typeset using exclusively T<sub>E</sub>X (L<sup>A</sup>T<sub>E</sub>X) and the *Al Amal* system; this paper has been printed on a Apple LaserWriter with Accel-A-Writer board, at 600 dpi.

## References

- [1] T. V. Raman, *An Audio View of T<sub>E</sub>X Documents*, Proceedings of the 13th Annual T<sub>E</sub>X Users Group Meeting, Portland, 1992.
- [2] Daniel Berry and Shimon Yanai, *Environment for translating METAFONT to PostScript*, TUGboat 11 (1990), No. 4.
- [3] القاسمي العلوي عزيز، عناصر الإحصاء الوصفي، سلسلة التعريب والمعرفة، ١٩٨٦، El Kacimi Alaoui Aziz, *Éléments de statistique descriptive*, Collection Arabisation et Connaissances, 1986.
- [4] Carl Faulmann, *Das Buch der Schrift, enthaltend die Schriftzeichen und Alphabete aller Zeiten und aller Völker des Erdkreises*, Verlag der Kaiserlich-Königlichen Hof- und Staatsdruckerei, Vienna, 1880.
- [5] Hacki Tewfik Galandjizade, *Türkisch-Deutsches Wörterbuch*, Otto Holtze's Nachfolger, Leipzig, 1907.
- [6] Yannis Haralambous, *Arabic, Persian and Ottoman T<sub>E</sub>X for Mac and PC*, TUGboat 11 (1990), No. 4.
- [7] Yannis Haralambous, *T<sub>E</sub>X and those other languages...*, Proceedings of the 1991 T<sub>E</sub>X Conference in Dedham, TUGboat 12 (1991), No. 3.
- [8] Yannis Haralambous, *Towards the revival of traditional Arabic typography... through T<sub>E</sub>X*, Proceedings of the 7th European T<sub>E</sub>X Conference, Prague 1992.
- [9] Yannis Haralambous, *L'arabe moderne à la portée de tout le monde...*, to appear in Cahiers GUTenberg.
- [10] هاشم محمد الخطاط، قواعد الخط العربي، عالم الكتب، بيروت ١٩٨٤
- [11] John D. Hobby, *A METAFONT-like System with PostScript Output*, Proceedings of the 1989 T<sub>E</sub>X Conference in Stanford, TUGboat 10 (1989), No. 4.
- [12] John D. Hobby, *Introduction to MetaPost*, Proceedings of the 7th European T<sub>E</sub>X Conference, Prague 1992.
- [13] D. E. Knuth, *The T<sub>E</sub>Xbook*, Computers and Typesetting, Volume A, Addison-Wesley, Reading, 1989.
- [14] D. E. Knuth, *The METAFONTbook*, Computers and Typesetting, Volume C, Addison-Wesley, Reading, 1986.

- [15] Helmut Kopka, *LaTeX, eine Einführung*, 3rd edition, Addison-Wesley, München, 1992.
- [16] Helmut Kopka, *LaTeX, Erweiterungsmöglichkeiten*, 3rd edition, Addison-Wesley, München, 1991.
- [17] المكتبة الخضراء للأطفال، أطفال الغابة، دار المعارف، القاهرة، ١٩٨٩.
- [18] القرآن الكريم، جمعية الدعوة الاسلامية، الجماهيرية العربية الليبية الشعبية الاشتراكية، دار الكتاب اللبناني، بيروت، ١٩٨٠.
- [19] Hellmut Ritter, *Über einige Regeln, die beim Drucken mit arabischen Typen zu beachten sind*, Zeitschrift der Deutschen Morgenländischen Gesellschaft 100/2, pp. 577–580 (1951).
- [20] Syed Barakat Ahmad, *Introduction to Qur'ānic Script*, Curzon Press LTD, London, 1984.

أَيْنَمَا تَكُونُوا يُدْرِكُكُمْ الْمَوْتُ وَلَوْ كُنْتُمْ فِي بُرُوجٍ مُشِيدَةٍ

وَإِنْ تَصِبْهُمْ حَسَنَةً يَقُولُوا هَذِهِ مِنْ عِنْدِ اللَّهِ وَإِنْ تَصِبْهُمْ

سَيِّئَةً يَقُولُوا هَذِهِ مِنْ عِنْدِكَ قُلْ كُلٌّ مِنْ عِنْدِ اللَّهِ

فَمَالِ هَؤُلَاءِ الْقَوْمِ لَا يَكَادُونَ يَفْقَهُونَ حَدِيثًا ﴿٧٨﴾ مَا

أَصَابَكَ مِنْ حَسَنَةٍ فَمِنَ اللَّهِ وَمَا أَصَابَكَ مِنْ سَيِّئَةٍ فَمِنَ

نَفْسِكَ وَأَرْسَلْنَاكَ لِلنَّاسِ رَسُولًا وَكَفَى بِاللَّهِ شَهِيدًا ﴿٧٩﴾

مَنْ يُطِعِ الرَّسُولَ فَقَدْ أَطَاعَ اللَّهَ وَمَنْ تَوَلَّى فَمَا أَرْسَلْنَاكَ

عَلَيْهِمْ حَفِيفًا ﴿٨٠﴾ وَيَقُولُونَ طَاعَةٌ فَإِذَا بَرَزُوا مِنْ عِنْدِكَ

بَيَّتَ طَائِفَةٌ مِنْهُمْ غَيْرَ الَّذِي تَقُولُ وَاللَّهُ يَكْتُبُ مَا

يَبْيُتُونَ فَاغْرِضْ عَنْهُمْ وَقُلْ عَلَى اللَّهِ وَكَفَى بِاللَّهِ وَكِيلًا

﴿٨١﴾ أَفَلَا يَتَدَبَّرُونَ الْقُرْآنَ وَلَوْ كَانَ مِنْ عِنْدِ غَيْرِ اللَّهِ