Computer Standards & Interfaces 14 (1992) 3–12
North-Holland

A proposal for Vietnamese character encoding standards in a unified text processing framework

James Do [a], Ngô Thanh Nhàn [b] and Hoàng Nguyên [c]

[a] *Mentor Graphics Corp./IC Group, 2045 Hamilton Ave., San Jose, CA 95125, USA*
[b] *Linguistic String Project, New York University, 251 Mercer Street, New York, NY 10012, USA*
[c] *Printing Software Development Unit, Xerox Corp., 101 Continental Blvd., ESC1-306, El Segundo, CA 90245, USA*

Abstract

Do, J., N.T. Nhàn and H. Nguyên, A proposal for Vietnamese character encoding standards in a unified text processing framework, Computer Standards & Interfaces 14 (1992) 3–12.

A Vietnamese 8-bit character encoding standard is proposed that satisfies adequacy of character representation, and permits compatibility with existing software, adherence to current communications protocols, and fidelity of glyphic rendering within a broader text processing framework. The encoding approach is mixed, combining both separable and precomposed architectures. This unified code table also provides the basis for consistent integration into emerging 16- and 32-bit character encoding standards for multilingual computing.

Keywords. ISO DIS 10646; ISO 2022; ISO 4873, Unicode; Vietnamese character set.

Introduction

This is a review of the issues and requirements for Vietnamese character code sets, followed by a concrete proposal – as a contribution toward the effort of achieving a solution that is complete and consistent, but is also cognizant of the specific characteristics and requirements of Vietnamese-language computing and communications, in its current and future states.

It is a critical time to resolve this issue, due to the mounting needs for widespread and systematic introduction of computing into Viet Nam, and due to the efforts of the Unicode Consortium and the International Organization for Standardization (ISO) toward achieving 16- and 32-bit standards, respectively [9,5].

We consider at length the 8-bit scheme because of its pervasiveness in personal computers today – the main platforms for Vietnamese-language computing in Viet Nam and the rest of the world.

1. The Vietnamese alphabet

The Vietnamese alphabet consists of the *consonants*:

b, c, d, đ (d-bar), [f,] g, h, [j,] k, l, m, n, p, q,r, s, t, v, [w,] x, [z]

and the *vowels*:

a, a (a-breve), â (a-circumflex),
e, ê (e-circumflex), i,
o, ô (o-circumflex), o (o-horn),
u, u (u-horn), y.

Each vowel may be composed with any one of the following *tone marks:*

\` (grave), ′ (acute),
ˀ (dotless-question), ˜ (tilde),
̣ (dot-below)

We call an *orthographic unit* a consonant, a vowel or a tone mark. An *orthographic character* is either a consonant, a vowel, or a vowel composed with a tone mark. Thus, a tone mark is an orthographic unit, but not an orthographic character.

Table 1
ISO 646 International Reference Version (= ASCII)

	0	1	2	3	4	5	6	7
0	nul	dle	sp	0	@	P	`	p
1	soh	dc1	!	1	A	Q	a	q
2	stx	dc2	"	2	B	R	b	r
3	etx	dc3	#	3	C	S	c	s
4	eot	dc4	$	4	D	T	d	t
5	enq	nak	%	5	E	U	e	u
6	ack	syn	&	6	F	V	f	v
7	bel	etb	'	7	G	W	g	w
8	bs	can	(	8	H	X	h	x
9	ht	em	)	9	I	Y	i	y
A	lf	sub	*	:	J	Z	j	z
B	vt	esc	+	;	K	]	k	{
C	ff	fs	,	<	L	\	l	|
D	cr	gs	-	=	M	[	m	}
E	so	rs	.	>	N	^	n	~
F	si	us	/	?	O	_	o	del

Table 3
Proposed standard Vietnamese character encoding

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	nul	dle	sp	0	@	P	`	p			nbsp	`	´	ˀ	~	.
1	soh	dc1	!	1	A	Q	a	q			Ă	ă				
2	stx	dc2	"	2	B	R	b	r			Â	â				
3	etx	dc3	#	3	C	S	c	s			Đ	đ				
4	eot	dc4	$	4	D	T	d	t			Ê	ê				
5	enq	nak	%	5	E	U	e	u			Ô	ô				
6	ack	syn	&	6	F	V	f	v			Ơ	ơ				
7	bel	etb	'	7	G	W	g	w			Ư	ư				
8	bs	can	(	8	H	X	h	x								
9	ht	em	)	9	I	Y	i	y								
A	lf	sub	*	:	J	Z	j	z								
B	vt	esc	+	;	K	]	k	{								
C	ff	fs	,	<	L	\	l	|								
D	cr	gs	-	=	M	[	m	}								
E	so	rs	.	>	N	^	n	~								
F	si	us	/	?	O	_	o	del								

Table 2
Proposed standard Vietnamese code set (character and glyph encodings shown together)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	nul	dle	sp	0	@	P	`	p	Ặ	Ậ	nbsp	`	´	ˀ	~	.
1	Ừ	Ỳ	!	1	A	Q	a	q	À	È	Ă	ă	à	ằ	ầ	è
2	Ứ	Ý	"	2	B	R	b	r	Á	É	Ấ	â	á	ắ	ấ	é
3	Ử	Ỷ	#	3	C	S	c	s	Ả	Ẻ	Đ	đ	ả	ẳ	ẩ	ẻ
4	Ữ	Ỹ	$	4	D	T	d	t	Ã	Ẽ	Ê	ê	ã	ẵ	ẫ	ẽ
5	Ự	Ỵ	%	5	E	U	e	u	Ạ	Ẹ	Ô	ô	ạ	ặ	ậ	ẹ
6	Ệ	Ộ	&	6	F	V	f	v	Ì	Ò	Ơ	ơ	ề	ì	ò	ồ
7	bel	etb	'	7	G	W	g	w	Í	Ó	Ư	ư	ế	í	ó	ố
8	bs	can	(	8	H	X	h	x	Ỉ	Ỏ	Ằ	Ầ	ể	ỉ	ỏ	ổ
9	ht	em	)	9	I	Y	i	y	Ĩ	Õ	Ắ	Ấ	ễ	ĩ	õ	ỗ
A	lf	sub	*	:	J	Z	j	z	Ị	Ọ	Ẳ	Ẩ	ệ	ị	ọ	ộ
B	vt	esc	+	;	K	]	k	{	Ờ	Ù	Ẵ	Ẫ	ờ	ù	ừ	ỳ
C	ff	fs	,	<	L	\	l	|	Ớ	Ú	Ề	Ồ	ớ	ú	ứ	ý
D	cr	gs	-	=	M	[	m	}	Ở	Ủ	Ế	Ố	ở	ủ	ử	ỷ
E	so	rs	.	>	N	^	n	~	Ỡ	Ũ	Ể	Ổ	ỡ	ũ	ữ	ỹ
F	si	us	/	?	O	_	o	del	Ợ	Ụ	Ễ	Ỗ	ợ	ụ	ự	ỵ

Table 4
Proposed standard Vietnamese glyph encoding

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0				0	@	P	`	p	Ặ	Ậ		`	´	ˀ	~	.
1	Ừ	Ỳ	!	1	A	Q	a	q	À	È	Ă	ă	à	ằ	ầ	è
2	Ứ	Ý	"	2	B	R	b	r	Á	É	Â	â	á	ắ	ấ	é
3	Ử	Ỷ	#	3	C	S	c	s	Ả	Ẻ	Đ	đ	ả	ẳ	ẩ	ẻ
4	Ữ	Ỹ	$	4	D	T	d	t	Ã	Ẽ	Ê	ê	ã	ẵ	ẫ	ẽ
5	Ự	Ỵ	%	5	E	U	e	u	Ạ	Ẹ	Ô	ô	ạ	ặ	ậ	ẹ
6	Ệ	Ộ	&	6	F	V	f	v	Ì	Ò	Ơ	ơ	ề	ì	ò	ồ
7			'	7	G	W	g	w	Í	Ó	Ư	ư	ế	í	ó	ố
8			(	8	H	X	h	x	Ỉ	Ỏ	Ằ	Ầ	ể	ỉ	ỏ	ổ
9			)	9	I	Y	i	y	Ĩ	Õ	Ắ	Ấ	ễ	ĩ	õ	ỗ
A			*	:	J	Z	j	z	Ị	Ọ	Ẳ	Ẩ	ệ	ị	ọ	ộ
B			+	;	K	]	k	{	Ờ	Ù	Ẵ	Ẫ	ờ	ù	ừ	ỳ
C			,	<	L	\	l	|	Ớ	Ú	Ề	Ồ	ớ	ú	ứ	ý
D			-	=	M	[	m	}	Ở	Ủ	Ế	Ố	ở	ủ	ử	ỷ
E			.	>	N	^	n	~	Ỡ	Ũ	Ể	Ổ	ỡ	ũ	ữ	ỹ
F			/	?	O	_	o		Ợ	Ụ	Ễ	Ỗ	ợ	ụ	ự	ỵ

2. Current proposals

In comparison to the alphabet in the 7-bit ISO 646 International Reference Version standard (*Table 1*), a Vietnamese character code set requires either:

(1) 19 additional slots for the orthographic units in a *separable* architecture; or

(2) 134 additional slots for the orthographic characters in a *precomposed* architecture.

Either of these encoding architectures provides an *adequate* representation of Vietnamese characters. This adequacy requirement is necessary, but not sufficient, for Vietnamese-language computing.

Observe that in a precomposed architecture, *six extra slots* are required beyond the 128 already available in 8-bit space with the hexadecimal values 80 through FF. These six extra slots are at the core of the difficulty in achieving a satisfactory 8-bit Vietnamese encoding standard to date.

These proposals and/or implementations exist for encoding Vietnamese:

(a) 8-bit precomposed: encode orthographic characters except for six 'rarely used' ones, preserve ISO 646-IRV [8];

Table 5
Proposed standard Vietnamese character names

	hex	PostScript® Name	Unicode®/10646 Name
ˋ	B0	(gravenonspacing)	NON-SPACING VIETNAMESE GRAVE
ˊ	C0	(acutenonspacing)	NON-SPACING VIETNAMESE ACUTE
’	D0	(dotlessquestionnonspacing)	NON-SPACING HOOK ABOVE
˜	E0	(tildenonspacing)	NON-SPACING TILDE
.	F0	(dotbelownonspacing)	NON-SPACING DOT BELOW
A	41	A	LATIN CAPITAL LETTER A
À	81	Agrave	LATIN CAPITAL LETTER A GRAVE
Á	82	Aacute	LATIN CAPITAL LETTER A ACUTE
Ả	83	(Adotlessquestion)	LATIN CAPITAL LETTER A HOOK ABOVE
Ã	84	Atilde	LATIN CAPITAL LETTER A TILDE
Ạ	85	(Adotbelow)	LATIN CAPITAL LETTER A DOT BELOW
Ă	A1	(Abreve)	LATIN CAPITAL LETTER A BREVE
Ằ	A8	(Abrevegrave)	LATIN CAPITAL LETTER A BREVE GRAVE
Ắ	A9	(Abreveacute)	LATIN CAPITAL LETTER A BREVE ACUTE
Ẳ	AA	(Abrevedotlessquestion)	LATIN CAPITAL LETTER A BREVE HOOK ABOVE
Ẵ	AB	(Abrevetilde)	LATIN CAPITAL LETTER A BREVE TILDE
Ặ	80	(Abrevedotbelow)	LATIN CAPITAL LETTER A BREVE DOT BELOW
Â	A2	Acircumflex	LATIN CAPITAL LETTER A CIRCUMFLEX
Ầ	B8	(Acircumflexgrave)	LATIN CAPITAL LETTER A CIRCUMFLEX GRAVE
Ấ	B9	(Acircumflexacute)	LATIN CAPITAL LETTER A CIRCUMFLEX ACUTE
Ẩ	BA	(Acircumflexdotlessquestion)	LATIN CAPITAL LETTER A CIRCUMFLEX HOOK ABOVE
Ẫ	BB	(Acircumflextilde)	LATIN CAPITAL LETTER A CIRCUMFLEX TILDE
Ậ	90	(Acircumflexdotbelow)	LATIN CAPITAL LETTER A CIRCUMFLEX DOT BELOW
B	42	B	LATIN CAPITAL LETTER B
C	43	C	LATIN CAPITAL LETTER C
D	44	D	LATIN CAPITAL LETTER D

(b) 8-bit precomposed: encode orthographic characters, replace several control (C0-space) and/or graphic characters in ISO 646-IRV [6,10];
(c) 8-bit separable: encode orthographic units, preserve ISO 646-IRV, ISO 8859/1 Latin-1 alphabet and C1-space;
(d) 16-bit separable (Unicode): encode orthographic units [9];
(e) 32-bit precomposed (ISO's DIS 10646): encode orthographic characters [5];

where C0-space with hexadecimal values 00 through 1F, and C1-space with 80 through 9F contain *control* characters (the other characters are known as *graphic* characters).

3. Encoding requirements

The 8-bit code sets listed above attempt to satisfy the following criteria:

Table 5 (continued)

Đ	A3	(Dbar)	LATIN CAPITAL D BAR
E	45	E	LATIN CAPITAL LETTER E
È	91	Egrave	LATIN CAPITAL LETTER E GRAVE
É	92	Eacute	LATIN CAPITAL LETTER E ACUTE
Ẻ	93	(Edotlessquestion)	LATIN CAPITAL LETTER E HOOK ABOVE
Ẽ	94	(Etilde)	LATIN CAPITAL LETTER E TILDE
Ẹ	95	(Edotbelow)	LATIN CAPITAL LETTER E DOT BELOW
Ê	A4	Ecircumflex	LATIN CAPITAL LETTER E CIRCUMFLEX
Ề	AC	(Ecircumflexgrave)	LATIN CAPITAL LETTER E CIRCUMFLEX GRAVE
Ế	AD	(Ecircumflexacute)	LATIN CAPITAL LETTER E CIRCUMFLEX ACUTE
Ể	AE	(Ecircumflexdotlessquestion)	LATIN CAPITAL LETTER E CIRCUMFLEX HOOK ABOVE
Ễ	AF	(Ecircumflextilde)	LATIN CAPITAL LETTER E CIRCUMFLEX TILDE
Ệ	06	(Ecircumflexdotbelow)	LATIN CAPITAL LETTER E CIRCUMFLEX DOT BELOW
F	46	F	LATIN CAPITAL LETTER F
G	47	G	LATIN CAPITAL LETTER G
H	48	H	LATIN CAPITAL LETTER H
I	49	I	LATIN CAPITAL LETTER I
Ì	86	Igrave	LATIN CAPITAL LETTER I GRAVE
Í	87	Iacute	LATIN CAPITAL LETTER I ACUTE
Ỉ	88	(Idotlessquestion)	LATIN CAPITAL LETTER I HOOK ABOVE
Ĩ	89	(Itilde)	LATIN CAPITAL LETTER I TILDE
Ị	8A	(Idotbelow)	LATIN CAPITAL LETTER I DOT BELOW
J	4A	J	LATIN CAPITAL LETTER J
K	4B	K	LATIN CAPITAL LETTER K
L	4C	L	LATIN CAPITAL LETTER L
M	4D	M	LATIN CAPITAL LETTER M
N	4E	N	LATIN CAPITAL LETTER N
O	4F	O	LATIN CAPITAL LETTER O
Ò	96	Ograve	LATIN CAPITAL LETTER O GRAVE

(1) *Adequacy of representation*: to permit the representation, in a data stream, of every orthographic character as character(s) from the code set. As previously mentioned, either a separable or precomposed architecture will be adequate, if fully implemented;

(2) *Compatibility of processing*: to preserve – when the code set is used as the basis of a locale – the functionality of ASCII-based software. This necessitates that the ISO 646-IRV code space not be violated;

(3) *Fidelity of rendering*: to permit the faithful representation of orthographic characters – known as *glyphs* – on any output device. With current display technology, this requires that each orthographic character must be assigned a unique code value and descriptive name as pointers to the proper rendering function [3];

(4) *Data communications*: to maintain the validity of existing data communications and transmission schemes – mostly still per-

Table 5 (continued)

Ó	97	Oacute	LATIN CAPITAL LETTER O ACUTE
Ỏ	98	(Odotlessquestion)	LATIN CAPITAL LETTER O HOOK ABOVE
Õ	99	Otilde	LATIN CAPITAL LETTER O TILDE
Ọ	9A	(Odotbelow)	LATIN CAPITAL LETTER O DOT BELOW
Ô	A5	Ocircumflex	LATIN CAPITAL LETTER O CIRCUMFLEX
Ồ	BC	(Ocircumflexgrave)	LATIN CAPITAL LETTER O CIRCUMFLEX GRAVE
Ố	BD	(Ocircumflexacute)	LATIN CAPITAL LETTER O CIRCUMFLEX ACUTE
Ổ	BE	(Ocircumflexdotlessquestion)	LATIN CAPITAL LETTER O CIRCUMFLEX HOOK ABOVE
Ỗ	BF	(Ocircumflextilde)	LATIN CAPITAL LETTER O CIRCUMFLEX TILDE
Ộ	16	(Ocircumflexdotbelow)	LATIN CAPITAL LETTER O CIRCUMFLEX DOT BELOW
Ơ	A6	(Ohorn)	LATIN CAPITAL LETTER O HORN
Ờ	8B	(Ohorngrave)	LATIN CAPITAL LETTER O HORN GRAVE
Ớ	8C	(Ohornacute)	LATIN CAPITAL LETTER O HORN ACUTE
Ở	8D	(Ohorndotlessquestion)	LATIN CAPITAL LETTER O HORN HOOK ABOVE
Ỡ	8E	(Ohorntilde)	LATIN CAPITAL LETTER O HORN TILDE
Ợ	8F	(Ohorndotbelow)	LATIN CAPITAL LETTER O HORN DOT BELOW
P	50	P	LATIN CAPITAL LETTER P
Q	51	Q	LATIN CAPITAL LETTER Q
R	52	R	LATIN CAPITAL LETTER R
S	53	S	LATIN CAPITAL LETTER S
T	54	T	LATIN CAPITAL LETTER T
U	55	U	LATIN CAPITAL LETTER U
Ù	9B	Ugrave	LATIN CAPITAL LETTER U GRAVE
Ú	9C	Uacute	LATIN CAPITAL LETTER U ACUTE
Ủ	9D	(Udotlessquestion)	LATIN CAPITAL LETTER U HOOK ABOVE
Ũ	9E	(Utilde)	LATIN CAPITAL LETTER U TILDE
Ụ	9F	(Udotbelow)	LATIN CAPITAL LETTER U DOT BELOW
Ư	A7	(Uhorn)	LATIN CAPITAL LETTER U HORN
Ừ	01	(Uhorngrave)	LATIN CAPITAL LETTER U HORN GRAVE

formed with 7-bit lines. This requires that the control characters in C0- and C1-space be preserved, and that the code set be adaptable to the various layered international data transmission protocols as defined in ISO 2022, ISO 4873, Kermit, ... [4, 5a, 5b]

It is clear that a single 8-bit code set does not contain enough code values to satisfy all of the above necessary requirements for Vietnamese text processing. If two sets are needed, which set should embody which functionalities? A basis for a solution requires a broader perspective.

4. The broader context

The development of computer technology and its spreading use in many different language environments have exposed many fundamental characteristics not heretofore apparent. It is now rec-

Table 5 (continued)

Ứ	02	(Uhornacute)	LATIN CAPITAL LETTER U HORN ACUTE
Ử	03	(Uhorndotlessquestion)	LATIN CAPITAL LETTER U HORN HOOK ABOVE
Ữ	04	(Uhorntilde)	LATIN CAPITAL LETTER U HORN TILDE
Ự	05	(Uhorndotbelow)	LATIN CAPITAL LETTER U HORN DOT BELOW
V	56	V	LATIN CAPITAL LETTER V
W	57	W	LATIN CAPITAL LETTER W
X	58	X	LATIN CAPITAL LETTER X
Y	59	Y	LATIN CAPITAL LETTER Y
Ỳ	11	(Ygrave)	LATIN CAPITAL LETTER Y GRAVE
Ý	12	(Yacute)	LATIN CAPITAL LETTER Y ACUTE
Ỷ	13	(Ydotlessquestion)	LATIN CAPITAL LETTER Y HOOK ABOVE
Ỹ	14	(Ytilde)	LATIN CAPITAL LETTER Y TILDE
Ỵ	15	(Ydotbelow)	LATIN CAPITAL LETTER Y DOT BELOW
Z	5A	Z	LATIN CAPITAL LETTER Z
a	61	a	LATIN SMALL LETTER A
à	C1	agrave	LATIN SMALL LETTER A GRAVE
á	C2	aacute	LATIN SMALL LETTER A ACUTE
ả	C3	(adotlessquestion)	LATIN SMALL LETTER A HOOK ABOVE
ã	C4	atilde	LATIN SMALL LETTER A TILDE
ạ	C5	(adotbelow)	LATIN SMALL LETTER A DOT BELOW
ă	B1	(abreve)	LATIN SMALL LETTER A BREVE
ằ	D1	(abrevegrave)	LATIN SMALL LETTER A BREVE GRAVE
ắ	D2	(abreveacute)	LATIN SMALL LETTER A BREVE ACUTE
ẳ	D3	(abrevedotlessquestion)	LATIN SMALL LETTER A BREVE HOOK ABOVE
ẵ	D4	(abrevetilde)	LATIN SMALL LETTER A BREVE TILDE
ặ	D5	(abrevedotbelow)	LATIN SMALL LETTER A BREVE DOT BELOW
â	B2	acircumflex	LATIN SMALL LETTER A CIRCUMFLEX
ầ	E1	(acircumflexgrave)	LATIN SMALL LETTER A CIRCUMFLEX GRAVE
ấ	E2	(acircumflexacute)	LATIN SMALL LETTER A CIRCUMFLEX ACUTE

ognized that any computing system must satisfactorily support, at least for text processing, these fundamental functional units:
(1) input,
(2) output,
(3) processing,
(4) storage, and
(5) transmission,
each of which may embody a different data model [1,7]. Besides this 'horizontal' partitioning of functionality, a 'vertical' partitioning exists as well, particularly for communications within the well-defined OSI layers and protocols.

The criteria listed in the previous section fit within this framework.

Just as sorting has been recognized as a function which cannot be adequately supported in any encoding scheme (including English with ASCII) and has therefore been satisfactorily relegated to software applications and/or a locale definition, so too must each of the above functions be considered on its own merit and in conjunction with

Table 5 (continued)

ẩ	E3	(acircumflexdotlessquestion)	LATIN SMALL LETTER A CIRCUMFLEX HOOK ABOVE
ẫ	E4	(acircumflextilde)	LATIN SMALL LETTER A CIRCUMFLEX TILDE
ậ	E5	(acircumflexdotbelow)	LATIN SMALL LETTER A CIRCUMFLEX DOT BELOW
b	62	b	LATIN SMALL LETTER B
c	63	c	LATIN SMALL LETTER C
d	64	d	LATIN SMALL LETTER D
đ	B3	(dbar)	LATIN SMALL LETTER D BAR
e	65	e	LATIN SMALL LETTER E
è	F1	egrave	LATIN SMALL LETTER E GRAVE
é	F2	eacute	LATIN SMALL LETTER E ACUTE
ẻ	F3	(edotlessquestion)	LATIN SMALL LETTER E HOOK ABOVE
ẽ	F4	(etilde)	LATIN SMALL LETTER E TILDE
ẹ	F5	(edotbelow)	LATIN SMALL LETTER E DOT BELOW
ê	B4	ecircumflex	LATIN SMALL LETTER E CIRCUMFLEX
ề	C6	(ecircumflexgrave)	LATIN SMALL LETTER E CIRCUMFLEX GRAVE
ế	C7	(ecircumflexacute)	LATIN SMALL LETTER E CIRCUMFLEX ACUTE
ể	C8	(ecircumflexdotlessquestion)	LATIN SMALL LETTER E CIRCUMFLEX HOOK ABOVE
ễ	C9	(ecircumflextilde)	LATIN SMALL LETTER E CIRCUMFLEX TILDE
ệ	CA	(ecircumflexdotbelow)	LATIN SMALL LETTER E CIRCUMFLEX DOT BELOW
f	66	f	LATIN SMALL LETTER F
g	67	g	LATIN SMALL LETTER G
h	68	h	LATIN SMALL LETTER H
i	69	i	LATIN SMALL LETTER I
ì	D6	igrave	LATIN SMALL LETTER I GRAVE
í	D7	iacute	LATIN SMALL LETTER I ACUTE
ỉ	D8	(idotlessquestion)	LATIN SMALL LETTER I HOOK ABOVE
ĩ	D9	(itilde)	LATIN SMALL LETTER I TILDE
ị	DA	(idotbelow)	LATIN SMALL LETTER I DOT BELOW
j	6A	j	LATIN SMALL LETTER J

the others so that a consistent framework can be achieved to insure computing with high fidelity.

Another consequence of this sorting issue is that the assignment of any particular code value to a new character is of little importance, as long as necessary criteria are met.

(The input functionality is beyond the scope of this document, but will be properly dealt with separately. Issues that need to be considered are: keyboards, cursor functions, user interface,...).

5. A proposed solution

Given the above considerations, we therefore propose a unified and mixed code table (*Tables 2 and 4*) which combines a pair of 8-bit code sets as follows:

(1) *A character encoding set (Table 3)*, using a separable architecture, that preserves ISO 646-IRV as well as C1-space, and encodes the orthographic units. In a character pair

Table 5 (continued)

k	6B	k	LATIN SMALL LETTER K
l	6C	l	LATIN SMALL LETTER L
m	6D	m	LATIN SMALL LETTER M
n	6E	n	LATIN SMALL LETTER N
o	6F	o	LATIN SMALL LETTER O
ò	E6	ograve	LATIN SMALL LETTER O GRAVE
ó	E7	oacute	LATIN SMALL LETTER O ACUTE
ỏ	E8	(odotlessquestion)	LATIN SMALL LETTER O HOOK ABOVE
õ	E9	otilde	LATIN SMALL LETTER O TILDE
ọ	EA	(odotbelow)	LATIN SMALL LETTER O DOT BELOW
ô	B5	ocircumflex	LATIN SMALL LETTER O CIRCUMFLEX
ồ	F6	(ocircumflexgrave)	LATIN SMALL LETTER O CIRCUMFLEX GRAVE
ố	F7	(ocircumflexacute)	LATIN SMALL LETTER O CIRCUMFLEX ACUTE
ổ	F8	(ocircumflexdotlessquestion)	LATIN SMALL LETTER O CIRCUMFLEX HOOK ABOVE
ỗ	F9	(ocircumflextilde)	LATIN SMALL LETTER O CIRCUMFLEX TILDE
ộ	FA	(ocircumflexdotbelow)	LATIN SMALL LETTER O CIRCUMFLEX DOT BELOW
ơ	B6	(ohorn)	LATIN SMALL LETTER O HORN
ờ	CB	(ohorngrave)	LATIN SMALL LETTER O HORN GRAVE
ớ	CC	(ohornacute)	LATIN SMALL LETTER O HORN ACUTE
ở	CD	(ohorndotlessquestion)	LATIN SMALL LETTER O HORN HOOK ABOVE
ỡ	CE	(ohorntilde)	LATIN SMALL LETTER O HORN TILDE
ợ	CF	(ohorndotbelow)	LATIN SMALL LETTER O HORN DOT BELOW
p	70	p	LATIN SMALL LETTER P
q	71	q	LATIN SMALL LETTER Q
r	72	r	LATIN SMALL LETTER R
s	73	s	LATIN SMALL LETTER S
t	74	t	LATIN SMALL LETTER T
u	75	u	LATIN SMALL LETTER U
ù	DB	ugrave	LATIN SMALL LETTER U GRAVE

where a tone mark exists, the diacritic must follow the vowel. This code set is to be used for data storage and transmission; and

(2) *A glyph encoding set (Table 4)*, using a precomposed architecture, that preserves ISO 646-IRV and encodes the orthographic characters. This code set is to be used for output and rendering, and is accompanied by a corresponding table of character names.

Consistency between both code sets is maintained by keeping the orthographic units at the same code values. This is also consistent and appropriate with current digital rendering technology, such as PostScript [2,3].

The majority of the upper-case orthographic characters with tone marks are encoded in C0- and C1-space, so that this *simple rendering principle* holds:

In a linear text display (i.e. when the text baseline is straight), these orthographic characters can be faithfully rendered as a composition of the base character followed by a non-spacing tone mark, if the base characters and the tone marks are appropriately designed in the display font.

Necessarily, the above two code sets must also be accompanied by:

(3) *A locate definition* – paralleling one previously proposed in [6] – to properly handle sorting, conversion between the two sets, as well as other issues.

This proposal fulfills all four necessary criteria for Vietnamese-language computing, as previously mentioned.

The code sets – and a sample keyboard – are being implemented on a Macintosh computer and with PostScript font technology.

6. A unifying proposal

Consistent with existing architectures in Unicode and ISO's DIS 10646, and also with current

Table 5 (continued)

ú	DC	uacute	LATIN SMALL LETTER U ACUTE
ủ	DD	(udotlessquestion)	LATIN SMALL LETTER U HOOK ABOVE
ũ	DE	(utilde)	LATIN SMALL LETTER U TILDE
ụ	DF	(udotbelow)	LATIN SMALL LETTER U DOT BELOW
ư	B7	(uhorn)	LATIN SMALL LETTER U HORN
ừ	EB	(uhorngrave)	LATIN SMALL LETTER U HORN GRAVE
ứ	EC	(uhornacute)	LATIN SMALL LETTER U HORN ACUTE
ử	ED	(uhorndotlessquestion)	LATIN SMALL LETTER U HORN HOOK ABOVE
ữ	EE	(uhorntilde)	LATIN SMALL LETTER U HORN TILDE
ự	EF	(uhorndotbelow)	LATIN SMALL LETTER U HORN DOT BELOW
v	76	v	LATIN SMALL LETTER V
w	77	w	LATIN SMALL LETTER W
x	78	x	LATIN SMALL LETTER X
y	79	y	LATIN SMALL LETTER Y
ỳ	FB	(ygrave)	LATIN SMALL LETTER Y GRAVE
ý	FC	(yacute)	LATIN SMALL LETTER Y ACUTE
ỷ	FD	(ydotlessquestion)	LATIN SMALL LETTER Y HOOK ABOVE
ỹ	FE	(ytilde)	LATIN SMALL LETTER Y TILDE
ỵ	FF	(ydotbelow)	LATIN SMALL LETTER Y DOT BELOW
z	7A	z	LATIN SMALL LETTER Z

discussion and efforts for merging these two encodings, we propose that:

Vietnamese orthographic characters be encoded as part of Unicode's General Script Area (i.e. a component of the prospective 10646M's Base Multilingual Plane).

This will insure direct compatibility with this 8-bit code set proposal as well as with other existing code sets, and will provide a standard international depository for character definitions, in order to attain maximum consistency for Vietnamese character representation and portability for Vietnamese text processing.

Conclusion

A set of necessary and sufficient criteria for adequate Vietnamese text processing was presented which consistently integrates the existing encoding proposals. A unified code table was then derived combining a unified pair of 8-bit code sets. In the context of a broader computing framework, and of the efforts to achieve a common multilingual encoding standard, the proposed unified pair of comprehensive solution that is consistent with the needs of Vietnamese-language and multilingual computing and communications.

References

[1] G. Adams, An architectural overview of the ILA Multilingual Toolkit, version 2.0, ILA Technical Report 91-1 March 1991.

[2] Adobe Systems Inc, *PostScript Language Reference Manual [The Red Book]*, second ed., (Addison-Wesley, Reading, MA, 1990).

[3] J. Do, Re: Diacritics, orthography, type design, font format, *Linguist List* 2, (0328) (June 1991).

[4] C. Gianone, A Kermit protocol extension for international character sets, April 1990.

[5] International Organization for Standardization, Draft International Standard 10646, May 1991.

[5a] International Organization for Standardization, *ISO 2022: 7-bit and 8-bit coded character sets - Code extension techniques*, 1986.

[5b] International Organization for Standardization, *ISO 4873: 8-bit code for information interchange - Structure and rules for implementation*, 1986.

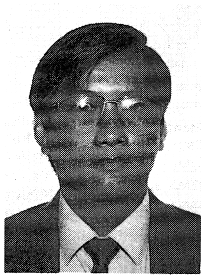
[6] T.V. Luong, VN.charmap and VN.locale, Viet-Std distribution list, June 1991.

[7] N.T. Nhàn, Vấn đề kiến trúc chữ' Việt trên máy tính điện tử [The Vietnamese coding problem for computers], *Đất Việt* (Nov. 1985).

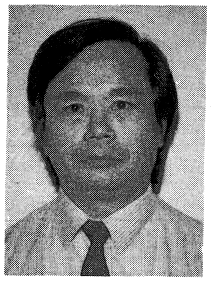
[8] Technical Group/Committee for Developing a Vietnamese Standard Code Set, A proposal for developing a Vietnamese standard code set (to be used in processing and communication between computers), Hanoi, 1987.

[9] Unicode Consortium, *The Unicode Standard: Worldwide Character Encoding* version 1-Pre-copy-edit draft, 7 May 1991.

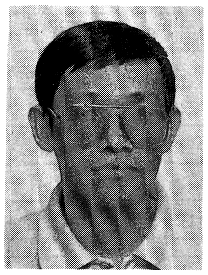
[10] J.W. van Wingen, Draft code table for Vietnamese letters for use on PC's, private communication, June 1991.



James Do is a software manager at Mentor Graphics Corporation, where he is involved in the development of electronic design automation software; specializing in software engineering issues, geometric optimization algorithms, and design language translation. He received an MA in Mathematics (1976) from the University of California, Irvine; and is a member of the Association for Computational Linguistics, the Association for Computing Machinery, the IEEE Computer Society, and the IFIP Working Group 6.4 'Social Implications of Computers in Developing Countries'. He has been interested in the design of fonts for the Vietnamese alphabet, and other issues related to Vietnamese computing and communications.



Ngô Thanh Nhàn, Linguistic String Project, New York University received a BA in Languages (1972) and an MA in linguistics (1977) at San Jose State University. He received a Ph. D. in Linguistics (1984) at NYU, specializing in Vietnamese word formation. In 1979–1984, he worked as an Assistant Research Scientist at NYU Computer Science Department, Courant Instute of Mathematical Sciences, first with the Question-Answering System and later on tuning natural language grammars for new domains. In 1984–1988, he worked as an Associate Research Scientist with the Proteus Project at NYU Computer Science Department, on automatic determination of sublanguage syntactic and selectional usage, with the sublanguage portability and robust parsing with graded acceptability, and later on model-based analysis of messages about equipments. Since 1988, he has worked on a joint project of NYU LSP and University Hospital of the Canton of Geneva Switzerland, on automatic encoding of clinical narratives, health care quality assessment and patient clinical profiles. During the years, Dr Nhàn has been interested in the subject of encoding of Vietnamese and *Nôm* traditional Vietnamese computational linguistics. He is a member of the Association for Computational Linguistics, and the Linguistics Society of Viet Nam.



Hoàng Nguyên is a member of the Printing Software Development Unit, Xerox Corporation with fourteen years of experience in system design, development, and integration. He received an MS in Computer Science (1977) from the University of California, Los Angeles; and is a member of the Association for Computing Machinery, and the IEEE Computer Society.