Workshop

Hinting

FDK

What is hinting?

The display of digital outlines on screen is a challenge, and often a problem. The issue is converting vector outlines to the pixel grid of the sceen.

Hinting is data in the font, which aims at helping with that. *Hinting* can also refer to the exciting activity of adding hints to a font.

Hinting data consists of various elements that may be font-wide settings, or applied to individual glyphs.

Most important hinting parameters:

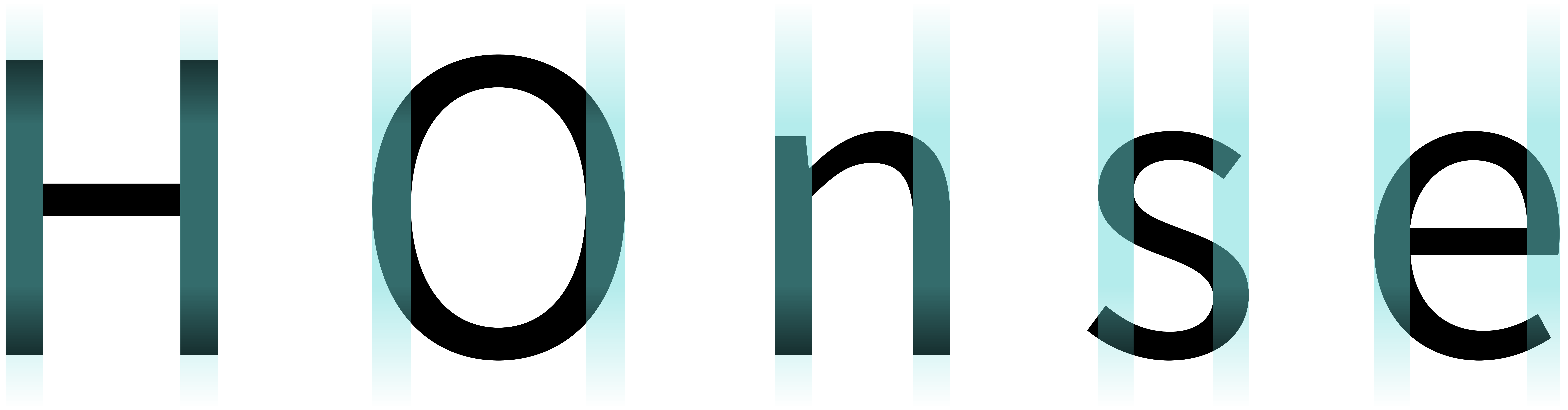
Zones and **Stems**

What is hinting?

CFF Hinting	TTF Hinting
cubic curves, counter-clockwise (T1 fonts, CFF-OTF, CFF-WOFF)	quadratic curves, clockwise (TTF, TTF-OTF, TTF-WOFF)
generalistic approach	glyph-by-glyph control
fine-grained control not possible	very fine-grained control
intelligence in rasterizer	intelligence in font
time spent on hinting: minutes	time spent on hinting: days
	better in older (Windows) environments

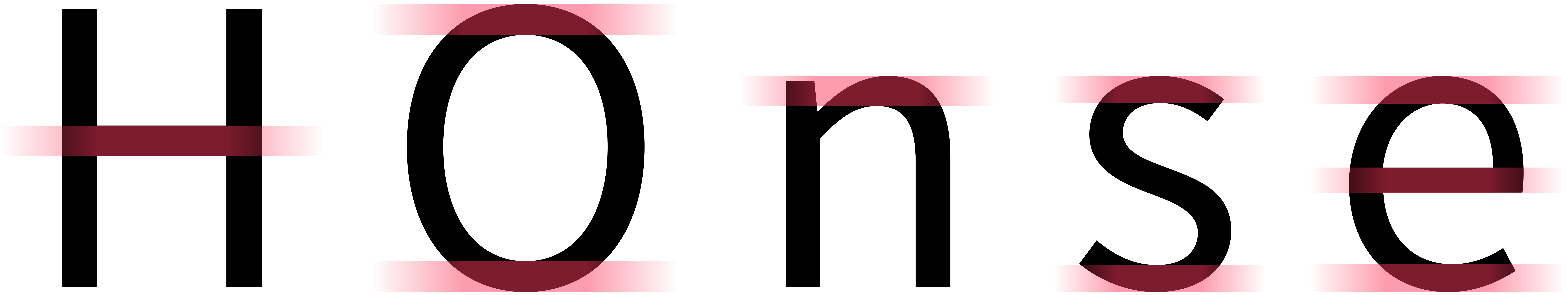
Hinting: Stems

Stems values tell the rasterizer which parts of the letter should have equal width. Vertical stems are also called *StemSnapV*.



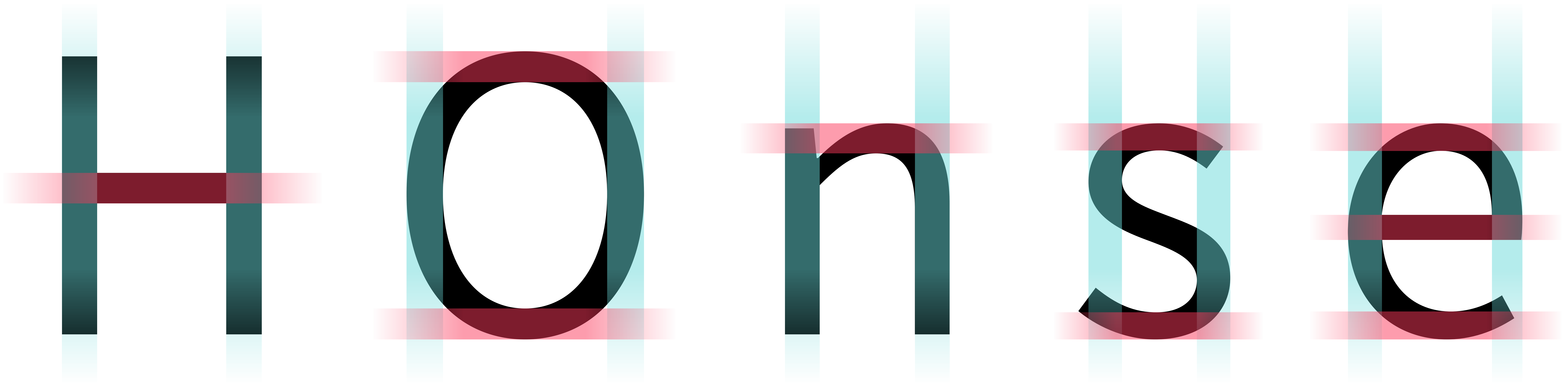
Hinting: Stems

Horizontal stems are also called *StemSnapH*.



Hinting: Stems

Keeping points at extremes, and consistent stem distances in the drawing are the best way to get a properly rasterizing font.



Hinting: Stems

Vertical stems (*StdVW*, *StemSnapV*)

– The first value is the overall dominant vertical stem width. Usually this value will match the width of the straight lowercase stems.

– The remaining values are the other most frequent vertical stem widths. Typically, these values will match the width of the straight stems of the uppercase letters

Horizontal stems (*StdHW*, *StemSnapH*)

– First value is the dominant horizontal stem width
– Remaining values are the other most frequent horizontal stem widths.

Hinting: Stems

- Stem values are a single list of numbers, with the dominant stem first, and the remaining stems sorted in increasing order (This means that the first value might be larger than the following numbers.)
- At least one value, and up to 12 values per direction must be specified.
- Amount and difference of stem values depend on the design.

Example StemSnapV for Source Sans Pro Regular: `84 95`

Example StemSnapH for Source Sans Pro Regular: `67 78`

FDK tools: stemHist

stemHist

A tool to help defining hinting parameters (stems).
Its result are two text files, that list all vertical and horizontal stems in order of occurrence. Can suggest alignment zones.

```
stemHist font.ufo
```

List all straight stems found, for all glyphs.

```
stemHist -all font.ufo
```

Also include round stems.

```
stemHist -g A-Z,a-z,zero-nine font.ufo
```

List stems for all glyphs between A and Z, between a and z, between zero and nine.

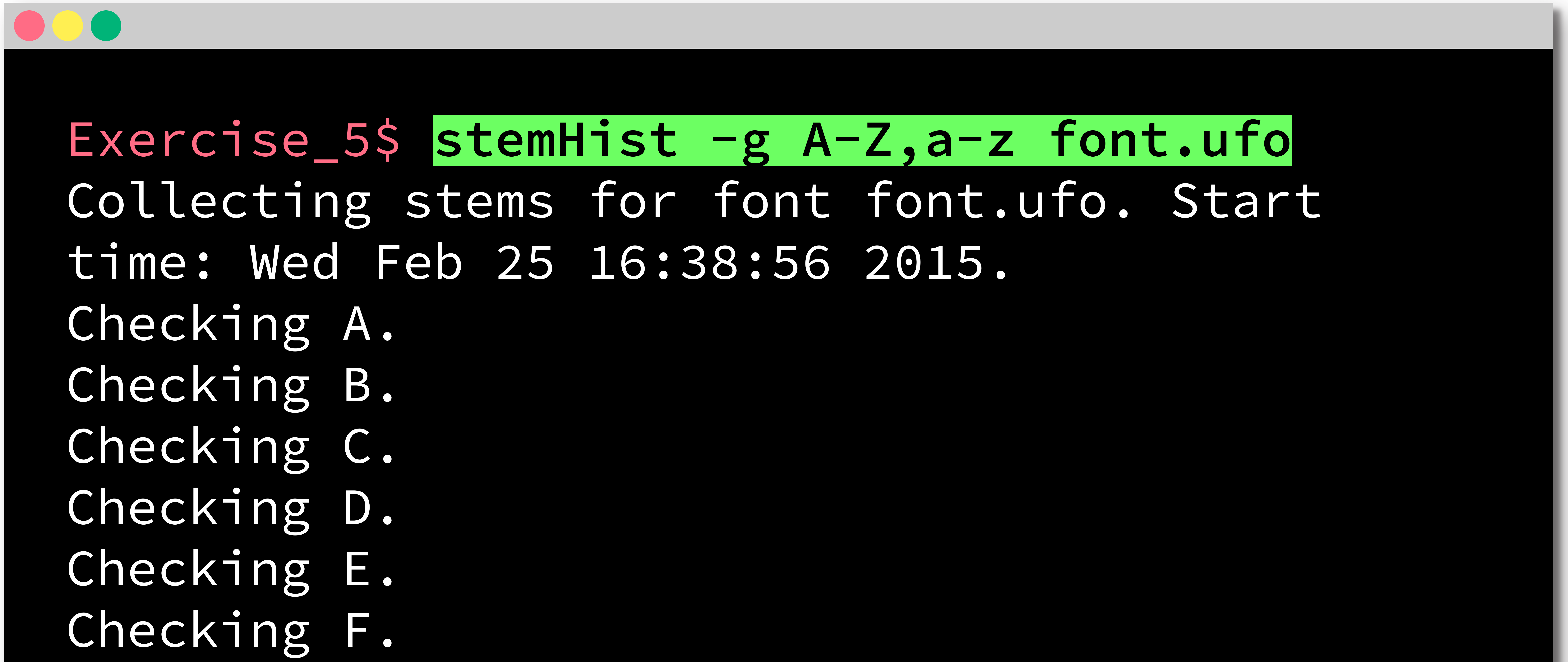
```
stemHist -a -g zero-nine font.ufo
```

List suggested alignment zone values for the figures of this font.

stemHist will run follow the internal glyph order of the font.

Exercise 5: Run stemHist!

In this case, we run stemHist only on the upper- and lowercase alphabet. We don’t need to redirect to a text file, two text files are created automatically.



```
Exercise_5$ stemHist -g A-Z,a-z font.ufo
Collecting stems for font font.ufo. Start
time: Wed Feb 25 16:38:56 2015.
Checking A.
Checking B.
Checking C.
Checking D.
Checking E.
Checking F.
```

Exercise 5: Analyze stemHist!

This is an example output for stemHist, we can analyze it together.

```
Vertical Stem List for font.ufo on Thu Mar 27 16:38:57 2014
Count  Width   Glyph List
15     88      ['a', 'd', 'g', 'h', 'i', 'j', 'l', 'm', 'n', 'p', 'q',]
11     87      ['M', 'N', 'b', 'f', 'h', 'k', 'm', 'n', 'r']
11     93      ['B', 'D', 'E', 'F', 'H', 'I', 'J', 'L', 'P', 'R', 'T']
3      94      ['H', 'K', 'Y']
1      388     ['z']
1      360     ['z']
1      404     ['Z']
1      441     ['Z']
1      90      ['G']
1      91      ['U']
1      92      ['U']
```

```
Horizontal Stem List for font.ufo on Thu Mar 27 16:38:57 2014
Count  Width   Glyph List
12     65      ['B', 'D', 'E', 'F', 'G', 'H', 'P', 'R', 'T', 'Z']
9      66      ['A', 'B', 'D', 'E', 'L', 'P', 'Z']
4      540     ['k', 'n', 'u', 'x']
4      59      ['f', 't', 'z']
4      738     ['d', 'h', 'k', 'l']
4      719     ['I', 'K', 'X', 'Y']
2      56      ['p', 'q']
2      70      ['M', 'W']
2      90      ['i', 'j']
1      48      ['g']
```

Hinting: Alignment Zones

Hinting: Alignment Zones

Hinting: Alignment Zones

Hinting: Alignment Zones

Hinting: Alignment Zones

Alignment Zones are narrow stripes, containing the overshoot of a particular glyph range.

Hinting: Alignment Zones

The zones at the top of glyphs are called *Blue Values*.

Note the exception: the baseline zone is also a Blue Value.

Hinting: Alignment Zones

The zones at the bottom of glyphs are called *Other Blues*.

Note the baseline zone of three.superior is also an Other Blue.

Hinting: Alignment Zones

Alignment Zones may shift and move across weights, but the amount of zones usually stays the same (they interpolate).

Hinting: Alignment Zones

Alignment Zones may shift and move across weights, but the amount of zones usually stays the same (they interpolate).

Hinting: Alignment Zones

FamilyBlues and *FamilyOtherBlues* equal the zone values of the Regular weight. *FamilyBlues* are used instead of *BlueValues* in any situation in which the difference is less than one pixel.

Blue Values of SSP-Black   FamilyBlues of SSP

Hinting: Alignment Zones

– Alignment Zones cannot overlap.
The minimum distance between consecutive zones is 1 unit.
(This is assuming that BlueFuzz is set to 0; more on that below.)

– Each zone is declared by a pair of integer values, in ascending order.

BlueValues, FamilyBlues:

– the first zone must be the baseline zone,
up to 7 zones may be defined

OtherBlues, FamilyOtherBlues:

– up to 5 zones may be defined

Hinting: Alignment Zones

Example BlueValues *for Source Sans Pro Regular:*

```
-12 0 486 498 518 530 574 586 638 650 656 668 712 724
```

-12	0:	Baseline overshoot
486	498:	x-height overshoot
518	530:	small caps overshoot
574	586:	old style figures overshoot
638	650:	lining figures overshoot
656	668:	cap overshoot
712	724:	ascender overshoot

Example OtherBlues *for Source Sans Pro Regular:*

-217	-205:	descender overshoot

Other hinting settings

BlueScale

– controls point size for overshoot to occur
– calculation of Medium BlueScale Value:

$$\frac{3}{4 \times \mathrm{MaxZoneSize}}$$

BlueShift

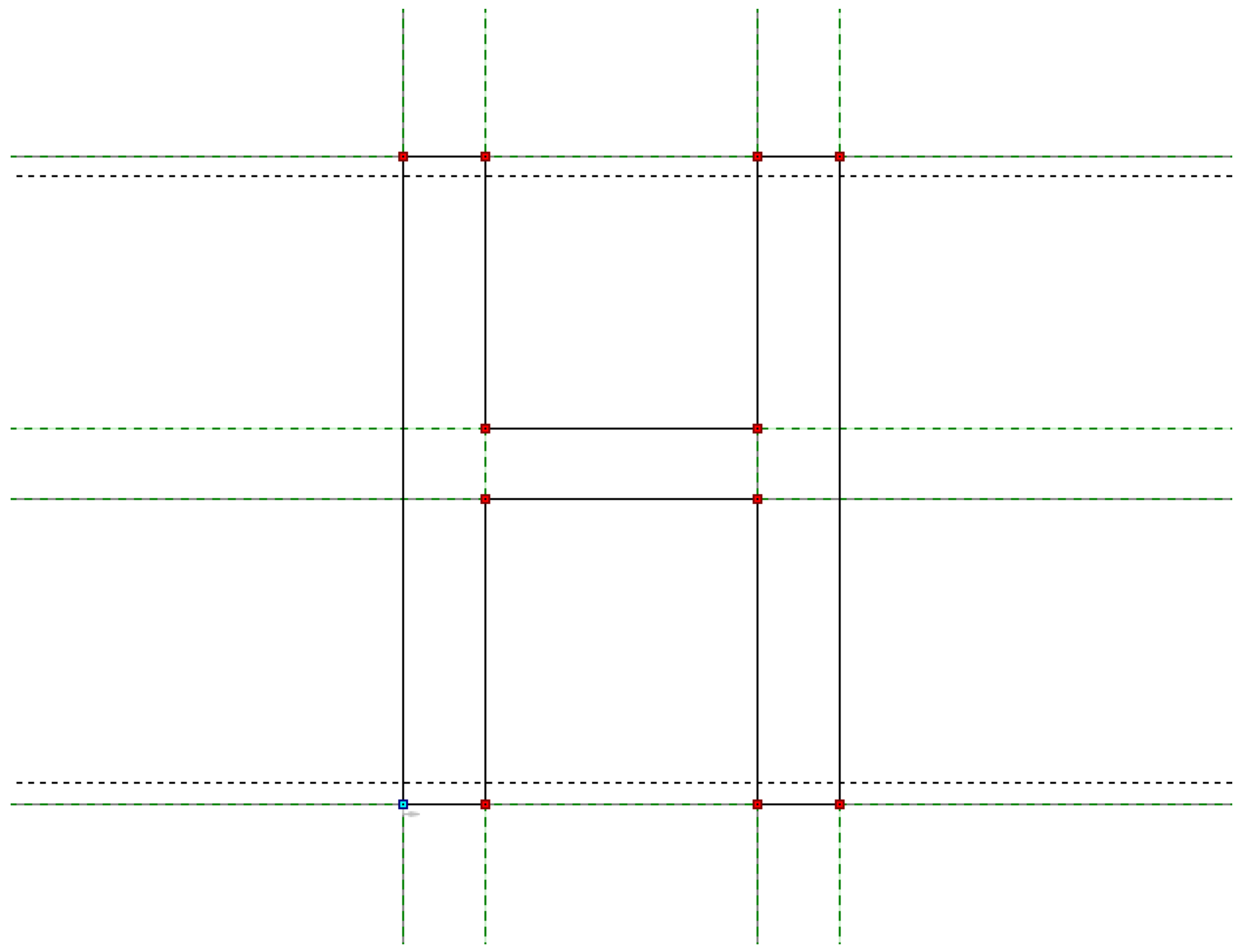
– Additional overshoot control.
  Defines the minimum overshoot distance to become visible.
  Default: 7 (@ 1000 UPM)

BlueFuzz

– Extends the zones in both directions.
  Don’t use it, set to 0.

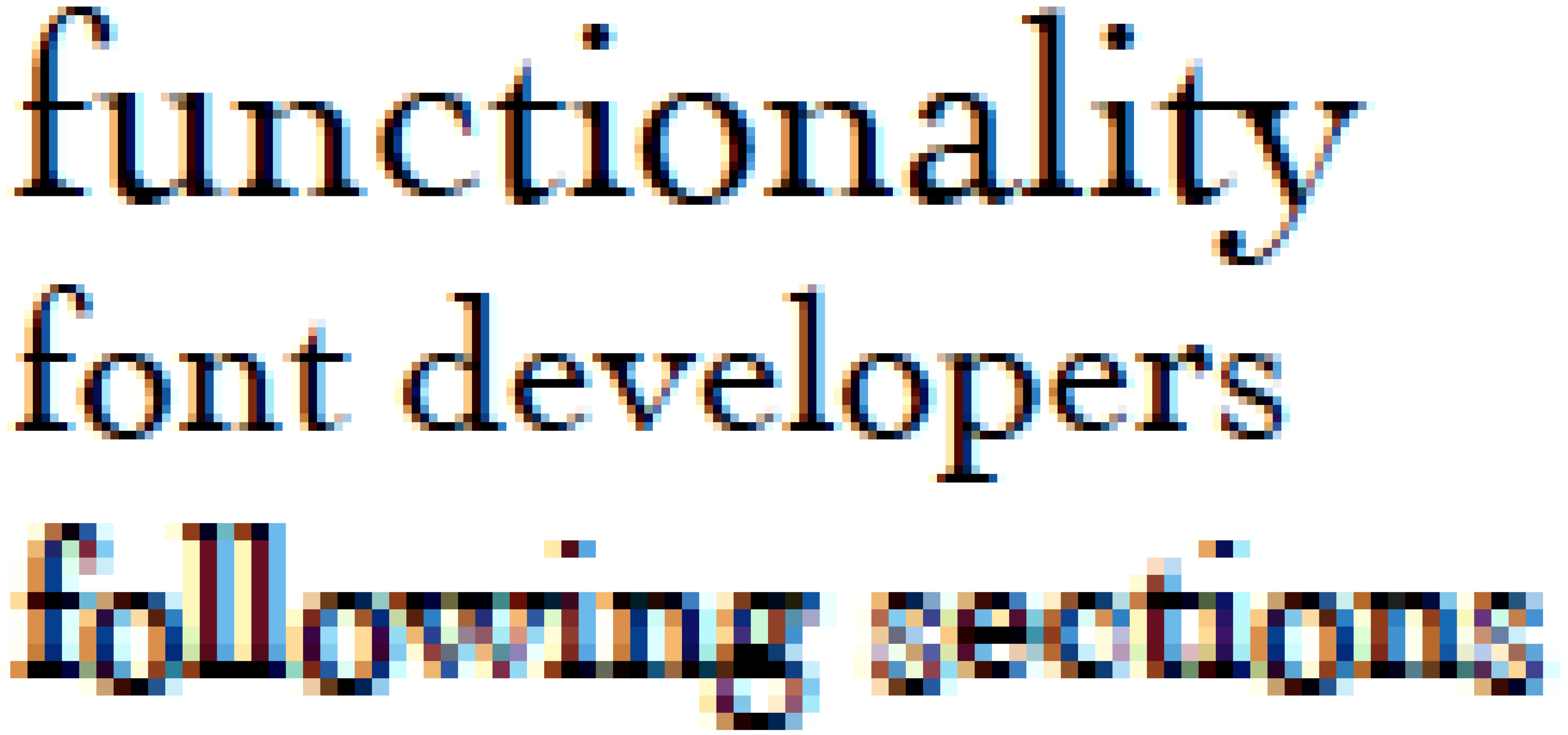
PostScript/CFF hinting

Spending some time on those those values will make your fonts look better. *Autohint* will be able to create hints that are very good, and don't need further attention.



Not hinting

“I don’t care about this stuff:”



This could have been avoided with just two alignment zones.

PostScript/CFF hinting

The effects of PostScript hinting depend on the rasterizer, and are most visible in Adobe Acrobat, or InDesign. In Apple Preview, you likely won’t see a difference, because it uses a different rasterizer.

In general, hinting will improve the display of type in smaller sizes. This does not mean that a font for large sizes needs no hinting – zooming out of a document will force a font into small display on screen, no matter what the intended usage size is.

BlueScale formulas

Minimum BlueScale:
(biggest zone equals half pixel)

$$\frac{1}{2 \times \text{MaxZoneSize}}$$

Medium BlueScale:
(value stored in font file)

$$\frac{3}{4 \times \text{MaxZoneSize}}$$

Maximum BlueScale:
(biggest zone equals whole pixel)

$$\frac{1}{\text{MaxZoneSize}}$$

BlueScale formulas

Overshoot threshold
(point size)

$$\frac{\text{BlueScale} \times 72 \times \text{UPM}}{\text{ppi}}$$

The overshoot threshold is the point size at which the overshoot starts happening.

ppi is the pixels per inch resolution of a specific rasterization environment (for instance, the MBP 15″ with Retina Display has 220 ppi). Rasterization however also occurs in most modern printing processes.

72 is a magic number, beause 1 inch is exactly 72 dtp points.

Exercise: Test overshoot settings in Acrobat

We can observe the effects of BlueScale values directly in Acrobat:

– autohint the PFA file

– create a waterfall PDF (low-res or Retina versions):

```
waterfallplot -wfr 48,49,50,51,52 -g x,o,x,o,x font.pfa
```

```
waterfallplot -wfr 23,24,25,26,27 -g x,o,x,o,x font.pfa
```

– open the PDF in Adobe Reader or Acrobat

– in the Preferences dialog, select the *Page Display* section.
Set the *Custom Resolution* to 72 pixels/inch.
Set *Smooth Text* to *For Laptop/LCD screens*

– set the zoom to 100%

What is TrueType Hinting?

TrueType Hinting is **not** magic!
Rather, it is the most boring thing on earth.
For TT Hinting, a font must be converted to TT outlines (clockwise, quadratic curves). Don’t even think about TT hinting while still in the design phase!

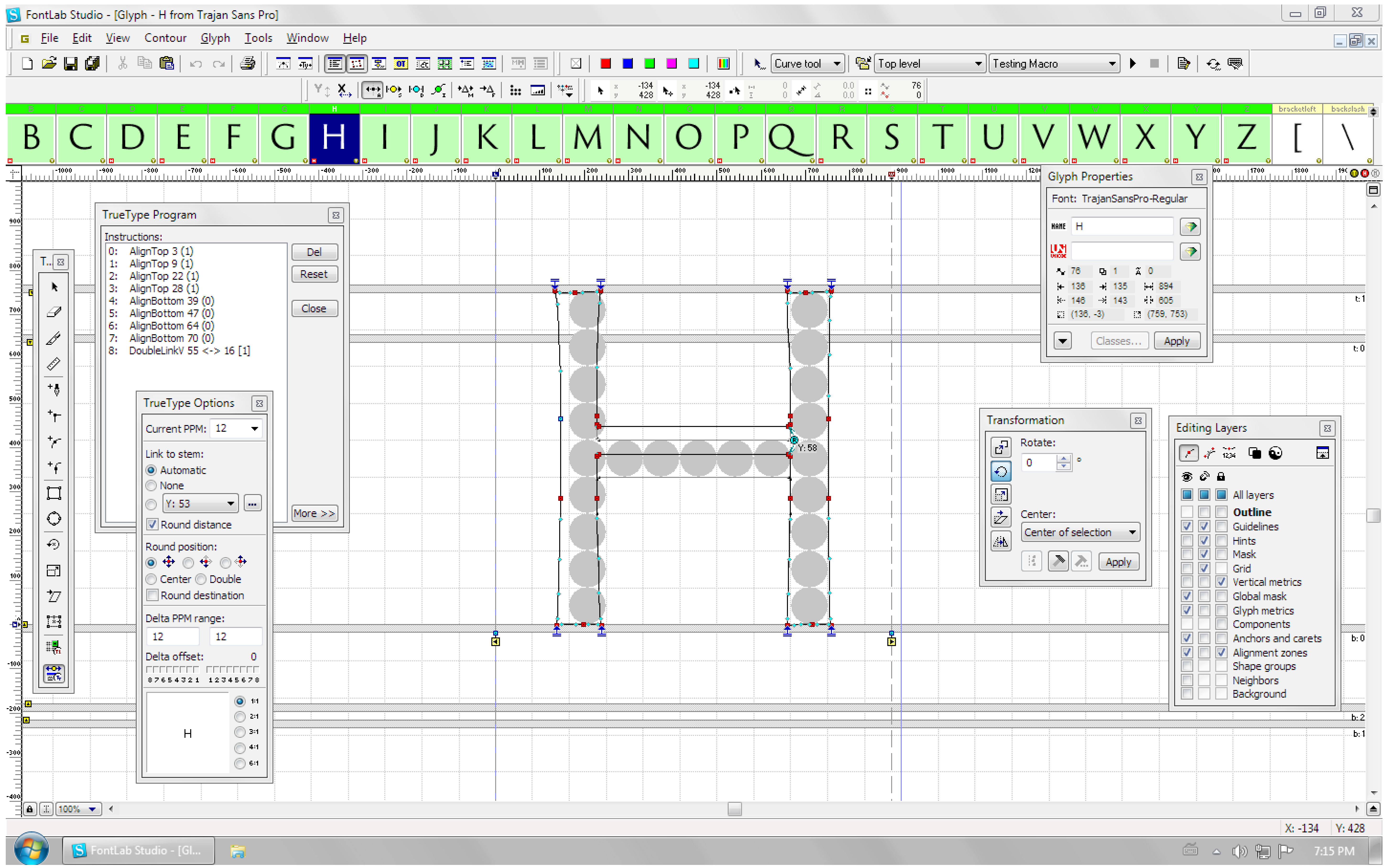
TT Hinting makes use of the following values:

Stems

Alignment Zones

PPMs

TrueType Hinting


FontLab Studio - [Glyph - H from Trajan Sans Pro]
File Edit View Contour Glyph Tools Window Help
Curve tool
Top level
Testing Macro
B C D E F G H I J K L M N O P Q R S T U V W X Y Z [\
Del
Reset
Close
More >>
Classes...
Apply
Y: 58
t: 1
t: 0
b: 0
b: 2
b: 1

TrueType Program

Instructions:

```
0:  AlignTop 3 (1)
1:  AlignTop 9 (1)
2:  AlignTop 22 (1)
3:  AlignTop 28 (1)
4:  AlignBottom 39 (0)
5:  AlignBottom 47 (0)
6:  AlignBottom 64 (0)
7:  AlignBottom 70 (0)
8:  DoubleLinkV 55 <-> 16 [1]
```

TrueType Options

Current PPM: 12

Link to stem: [✓ Automatic] None Y: 53

[✓ Round distance]

Round position: Center Double

Round destination

Delta PPM range: 12 12

Delta offset: 0

Glyph Properties

Font: TrajanSansPro-Regular

Name: H

76 1 0
136 135 894
146 143 605
(136, -3) (759, 753)

Transformation

Rotate: 0 °

Center: Center of selection

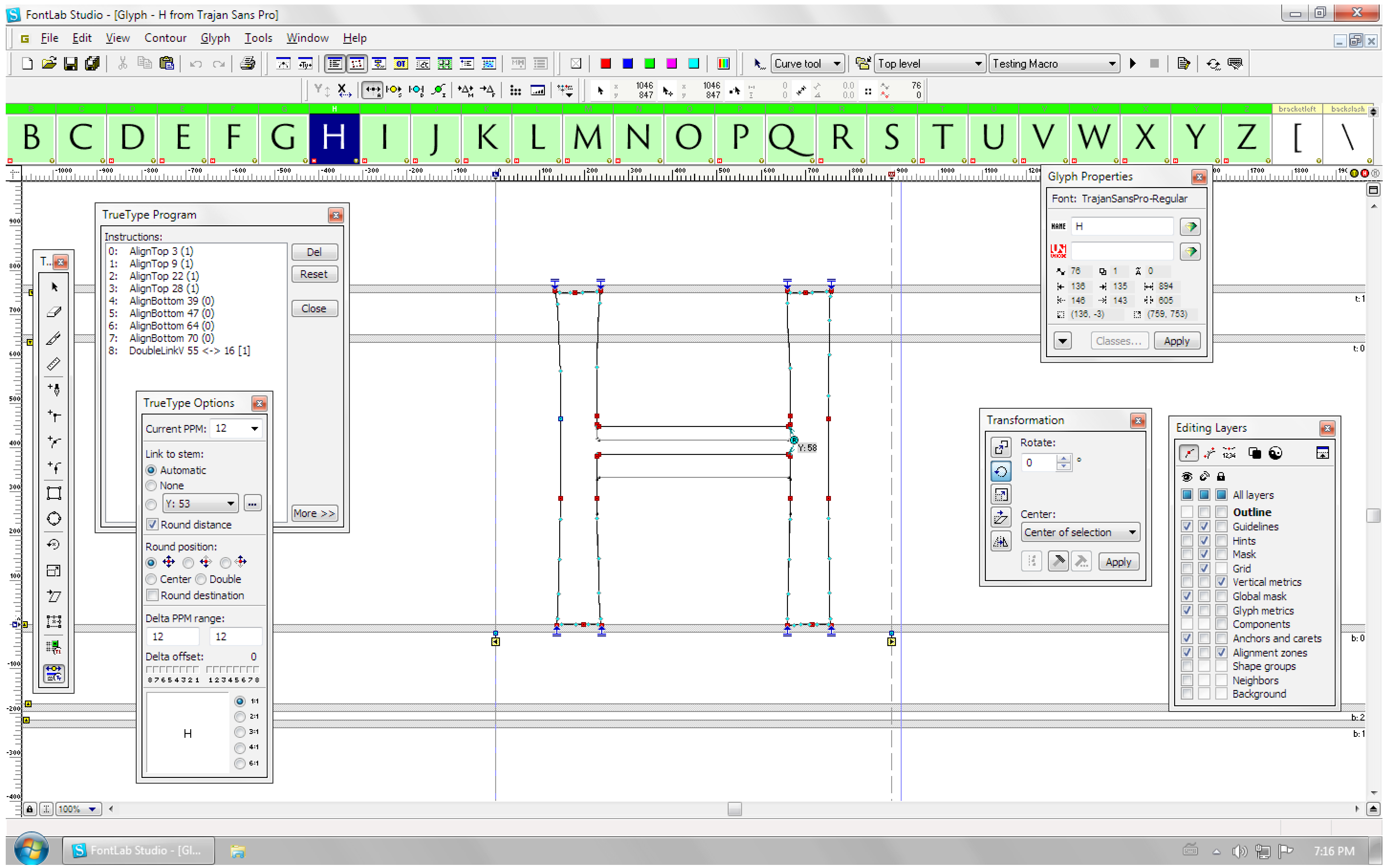
Editing Layers

All layers
Outline
[✓ Guidelines]
[✓ Hints]
[✓ Mask]
[✓ Grid]
[✓ Vertical metrics]
[✓ Global mask]
[✓ Glyph metrics]
Components
[✓ Anchors and carets]
[✓ Alignment zones]
Shape groups
Neighbors
Background

X: -134 Y: 428

7:15 PM

TrueType Hinting



FontLab Studio - [Glyph - H from Trajan Sans Pro]

TrueType Program

Instructions:

```
0:   AlignTop 3 (1)
1:   AlignTop 9 (1)
2:   AlignTop 22 (1)
3:   AlignTop 28 (1)
4:   AlignBottom 39 (0)
5:   AlignBottom 47 (0)
6:   AlignBottom 64 (0)
7:   AlignBottom 70 (0)
8:   DoubleLinkV 55 <-> 16 [1]
```

TrueType Options

Current PPM: 12

Link to stem: [✓ Automatic] None Y: 53

[✓ Round distance]

Round position:

Center Double

Round destination

Delta PPM range: 12 12

Delta offset: 0

[✓ 1:1] 2:1 3:1 4:1 6:1

Glyph Properties

Font: TrajanSansPro-Regular

NAME H

76 1 0
136 135 894
146 143 605
(136, -3) (759, 753)

Transformation

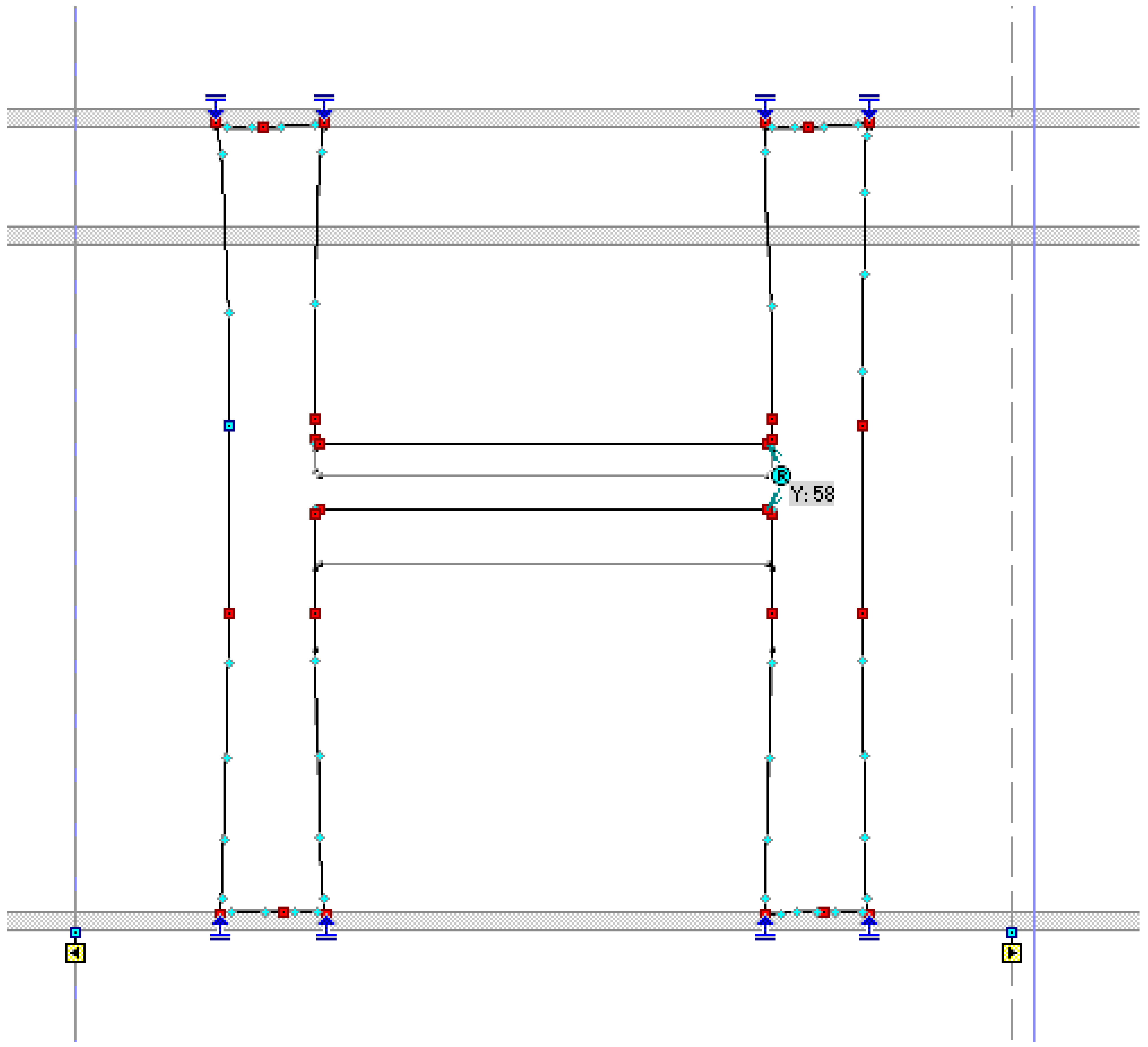
Rotate: 0 °

Center: Center of selection

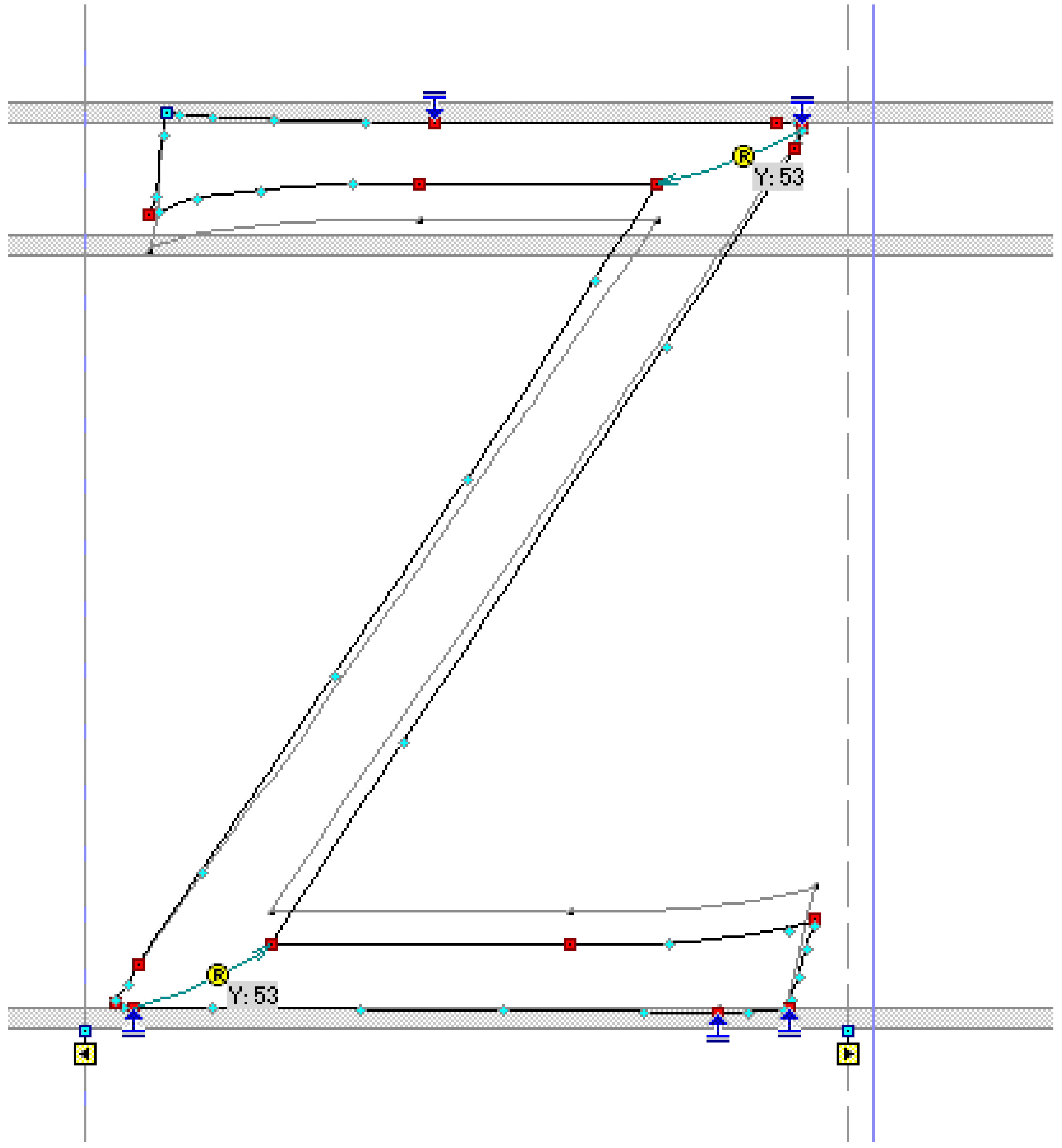
Editing Layers

			All layers
			Outline
✓	✓		Guidelines
	✓		Hints
	✓		Mask
	✓		Grid
		✓	Vertical metrics
✓			Global mask
✓			Glyph metrics
			Components
✓			Anchors and carets
✓		✓	Alignment zones
			Shape groups
			Neighbors
			Background

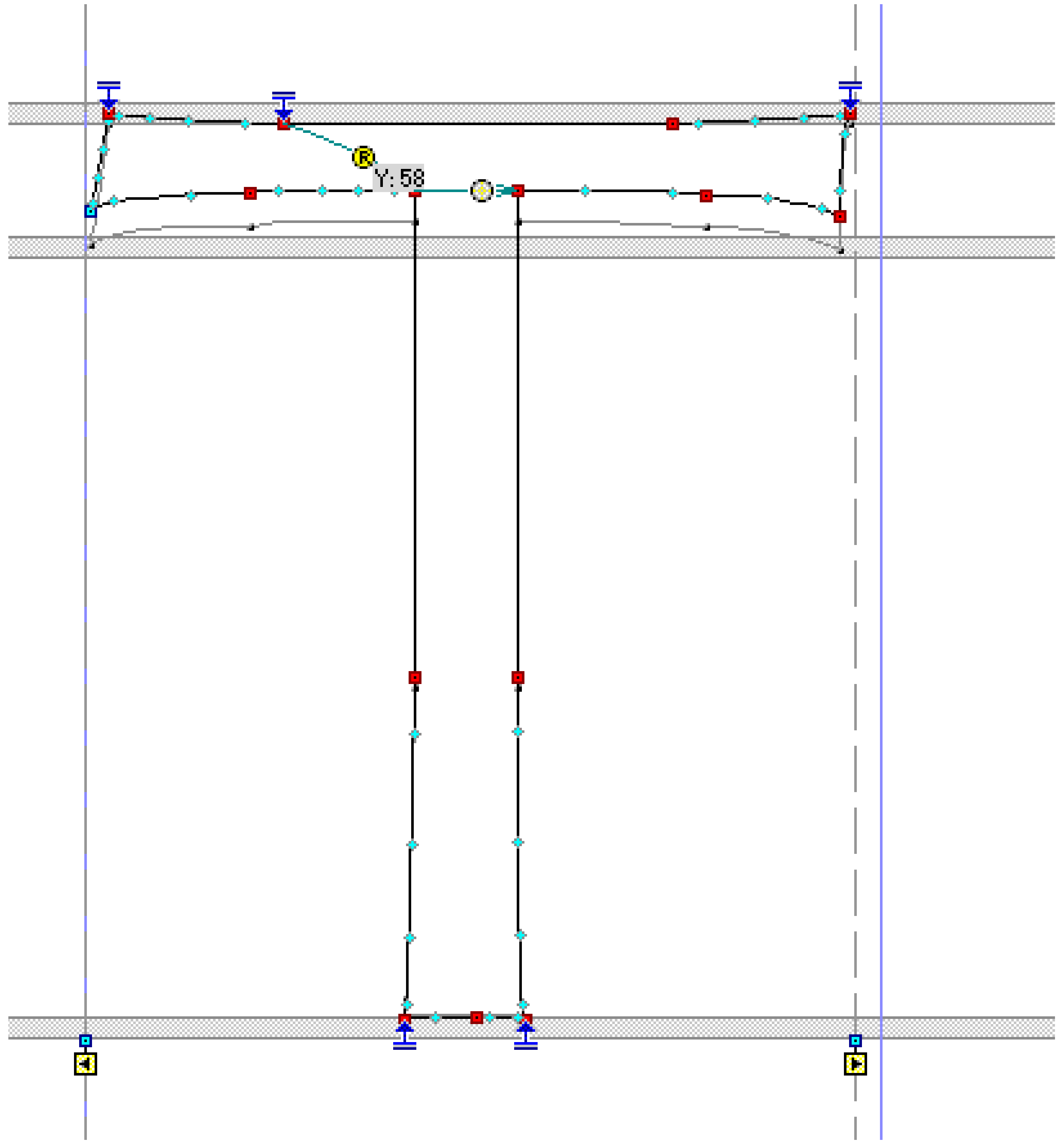
TrueType Hinting



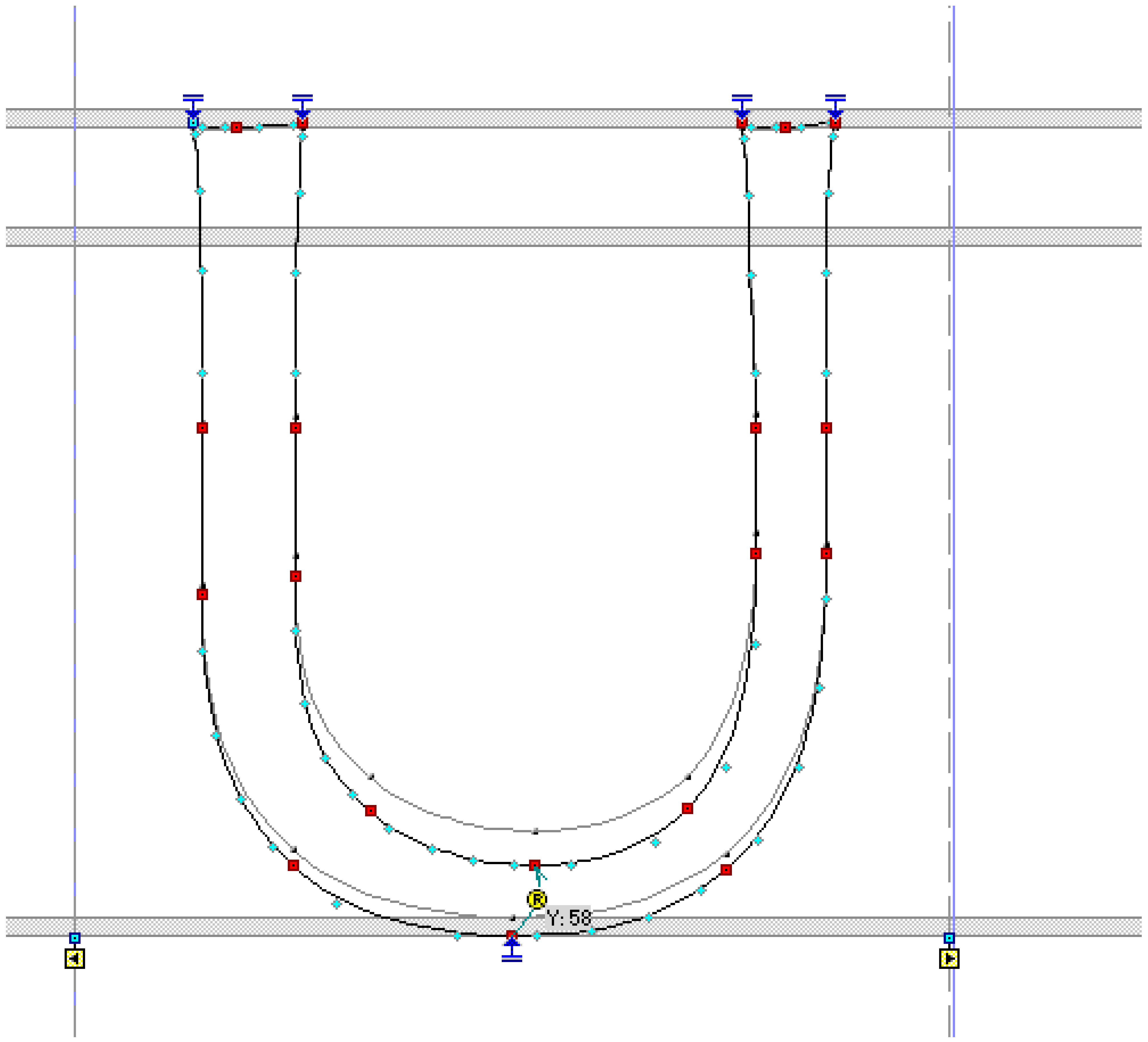
TrueType Hinting



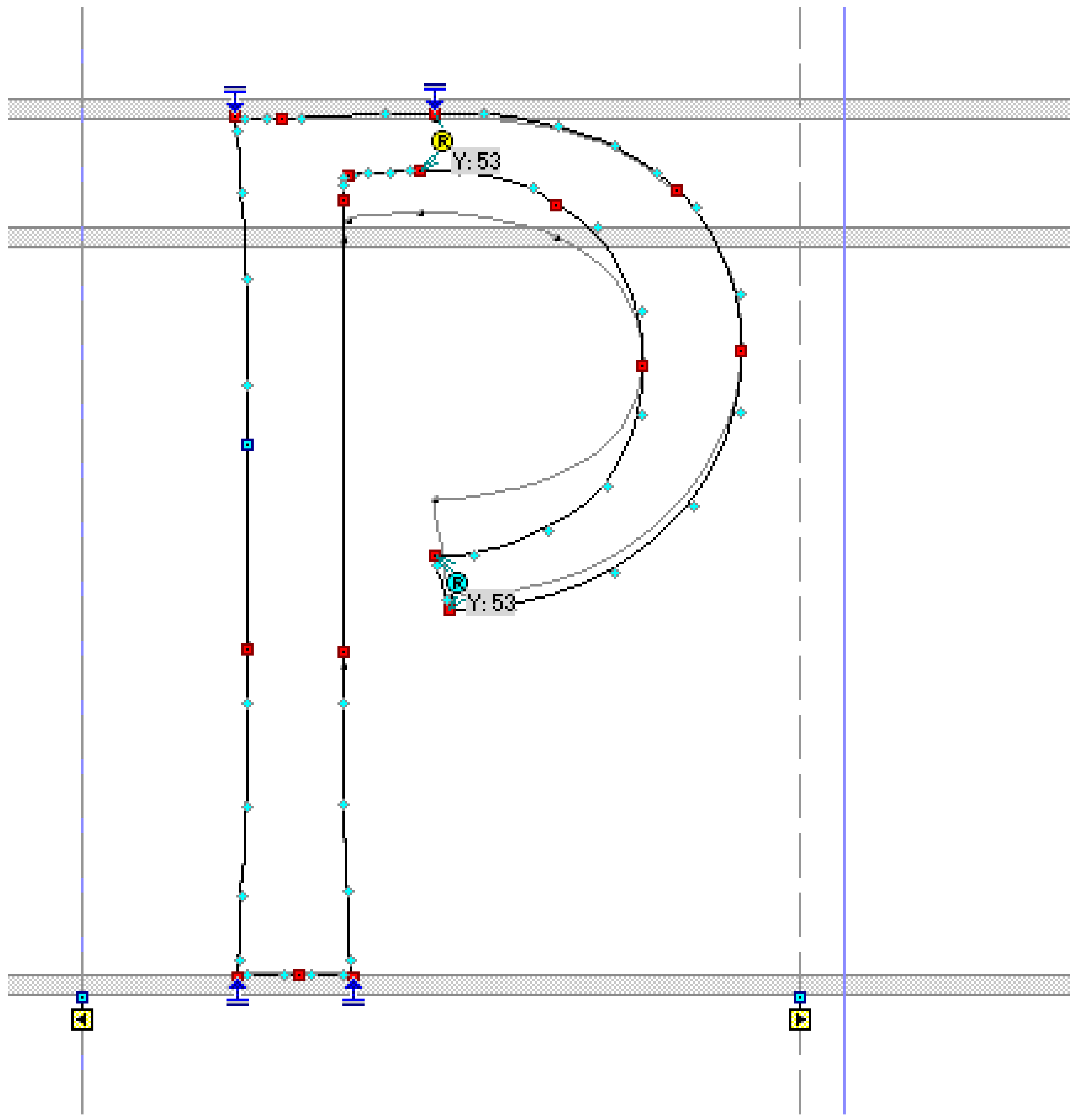
TrueType Hinting



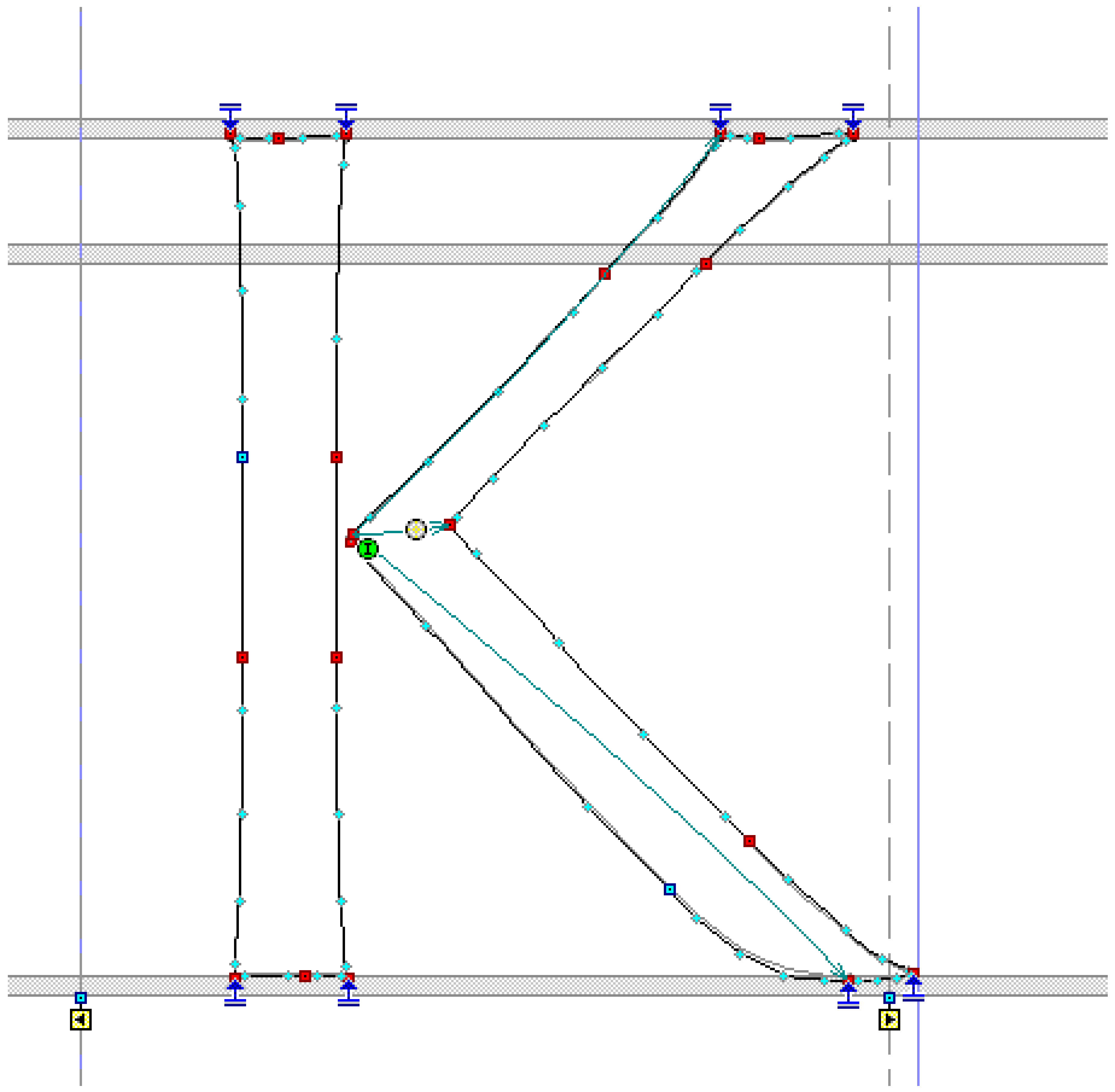
TrueType Hinting



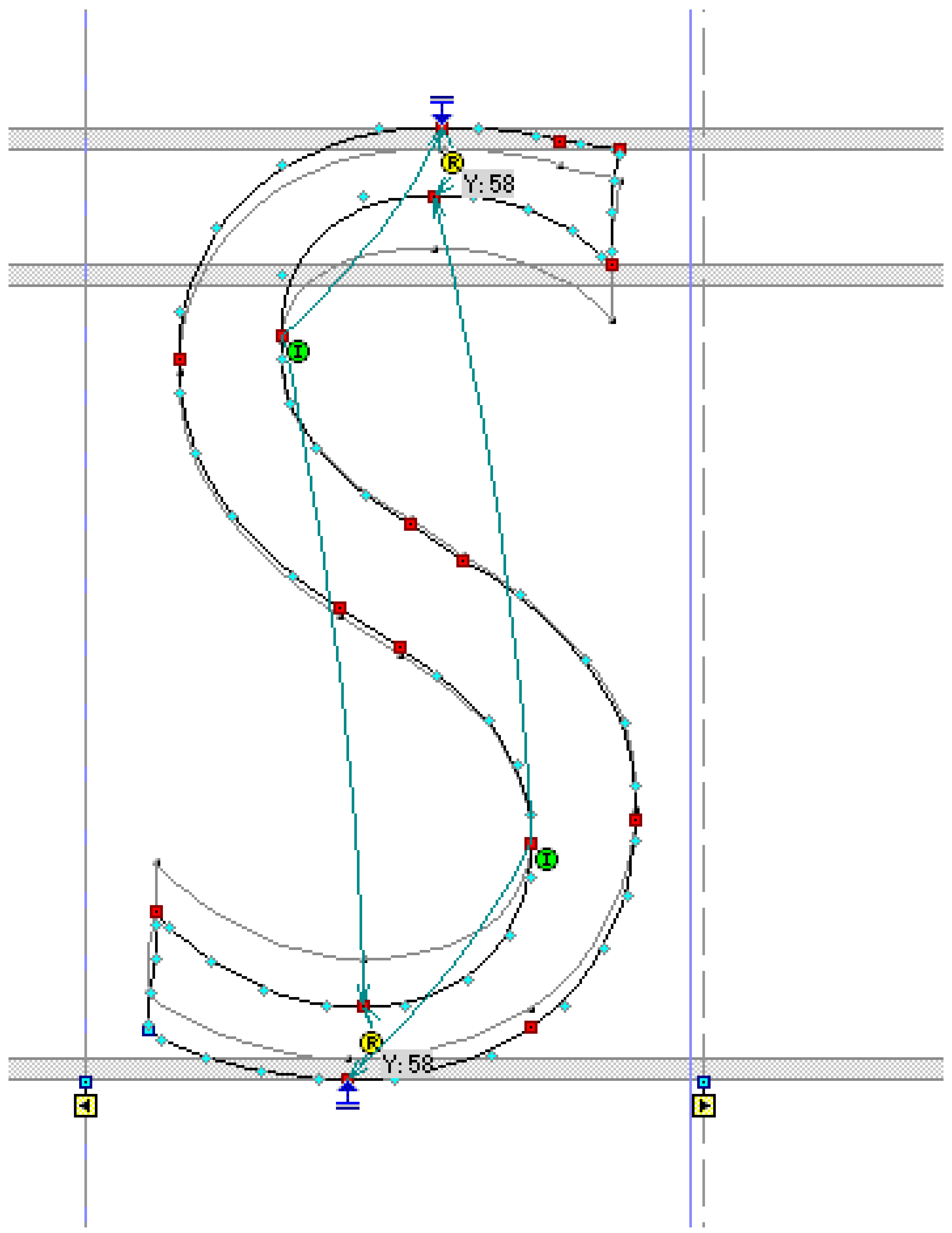
TrueType Hinting



TrueType Hinting



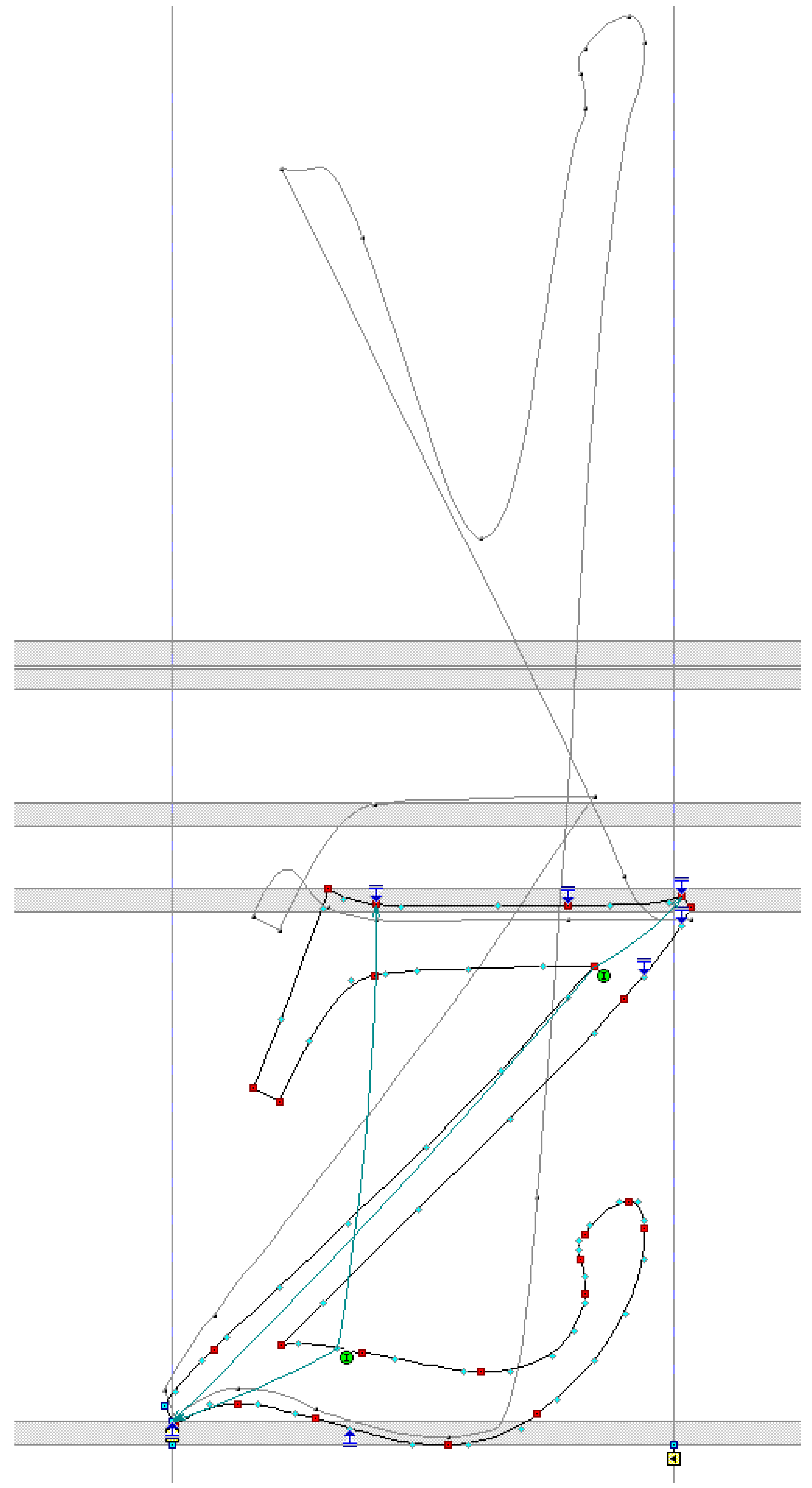
TrueType Hinting

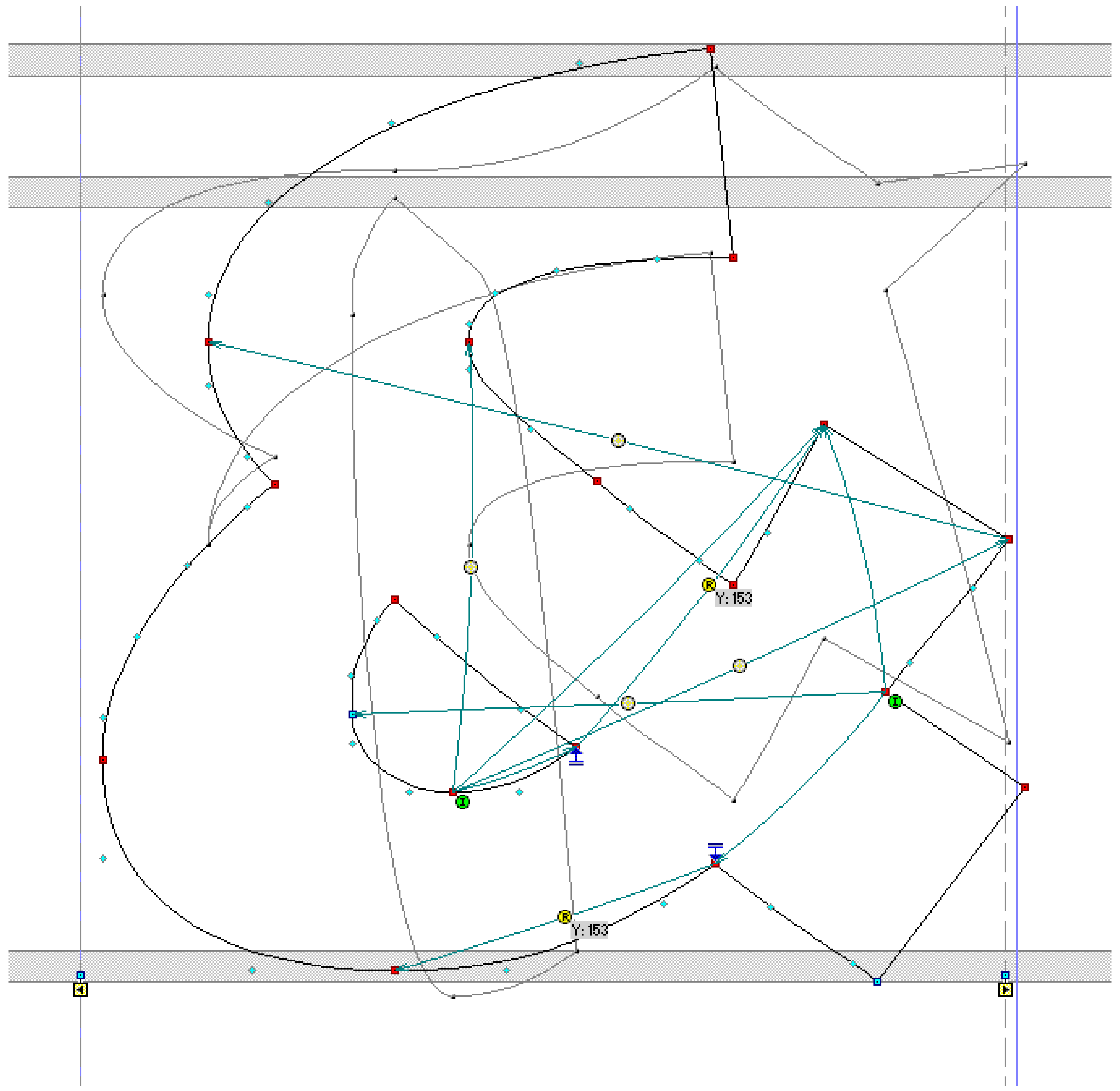


TrueType Hinting

anchor	Attaches a point to an alignment zone.
single link	Defines a stem width, must be attached to anchor or previous single-link, or interpolation.
double link	Defines a stem width without attachment to any kind of other hint.
interpolation	Connects three points; makes clear that the relation of the middle point to the outer two points is important.
delta	Can modify the position of a single point at one particular size. Avoid when possible.

TrueType Hinting gone bad!

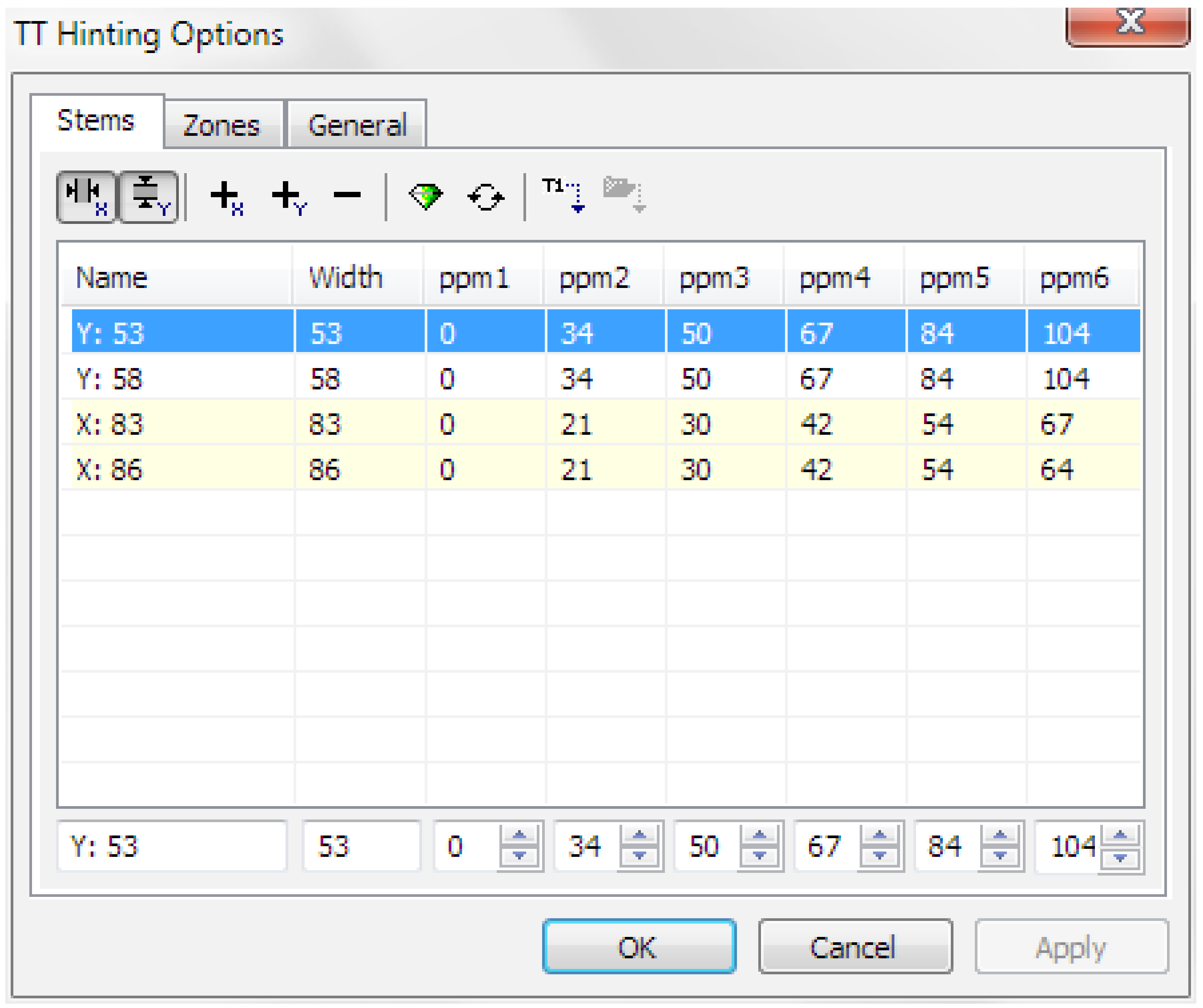




The superior level of possible control comes with a superior chance for possbile screw-up.

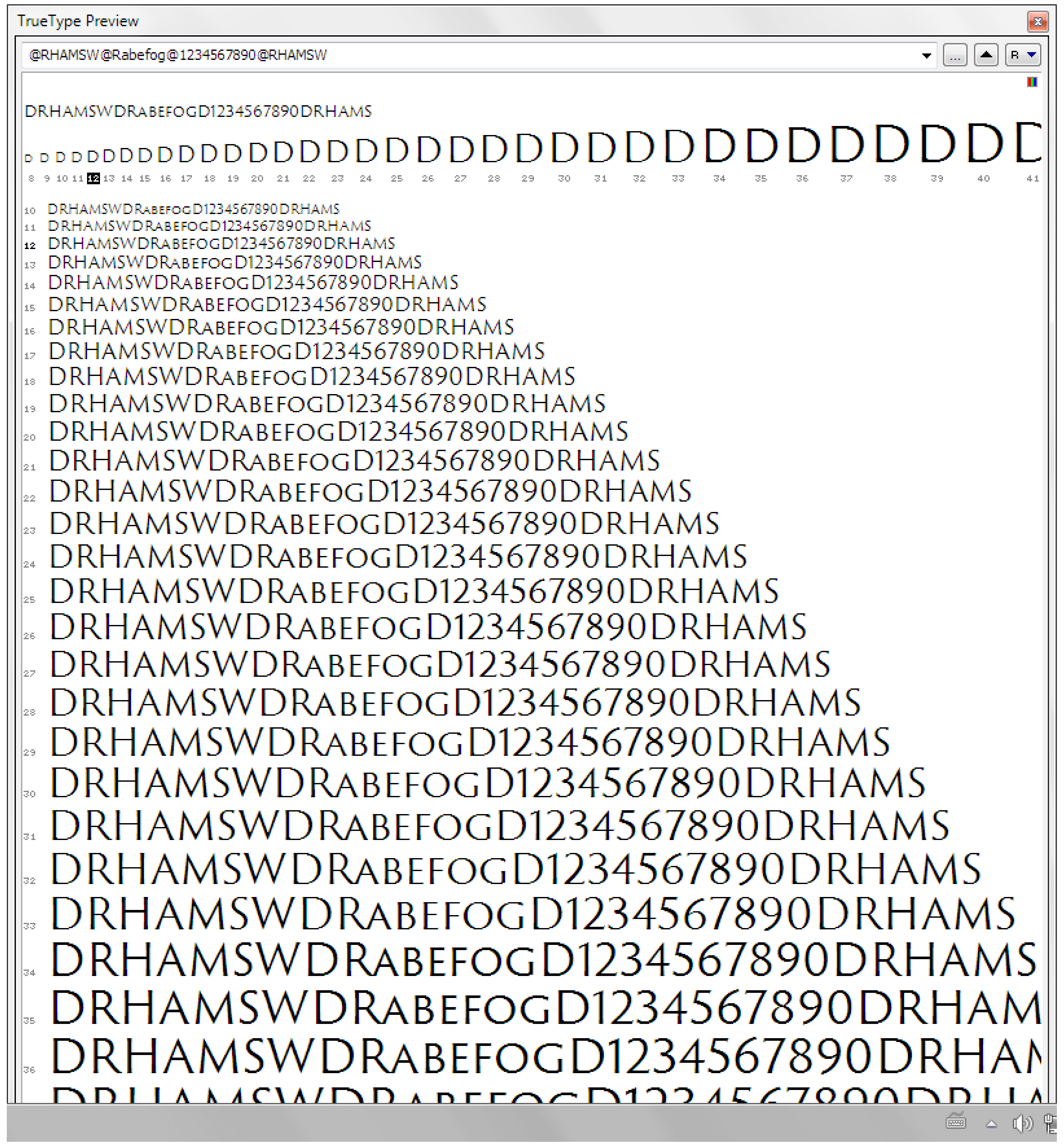
TrueType Hinting: PPM values

PPM values are responsible for the decision when a particular stem gets to be 2,3,4 … pixels wide.

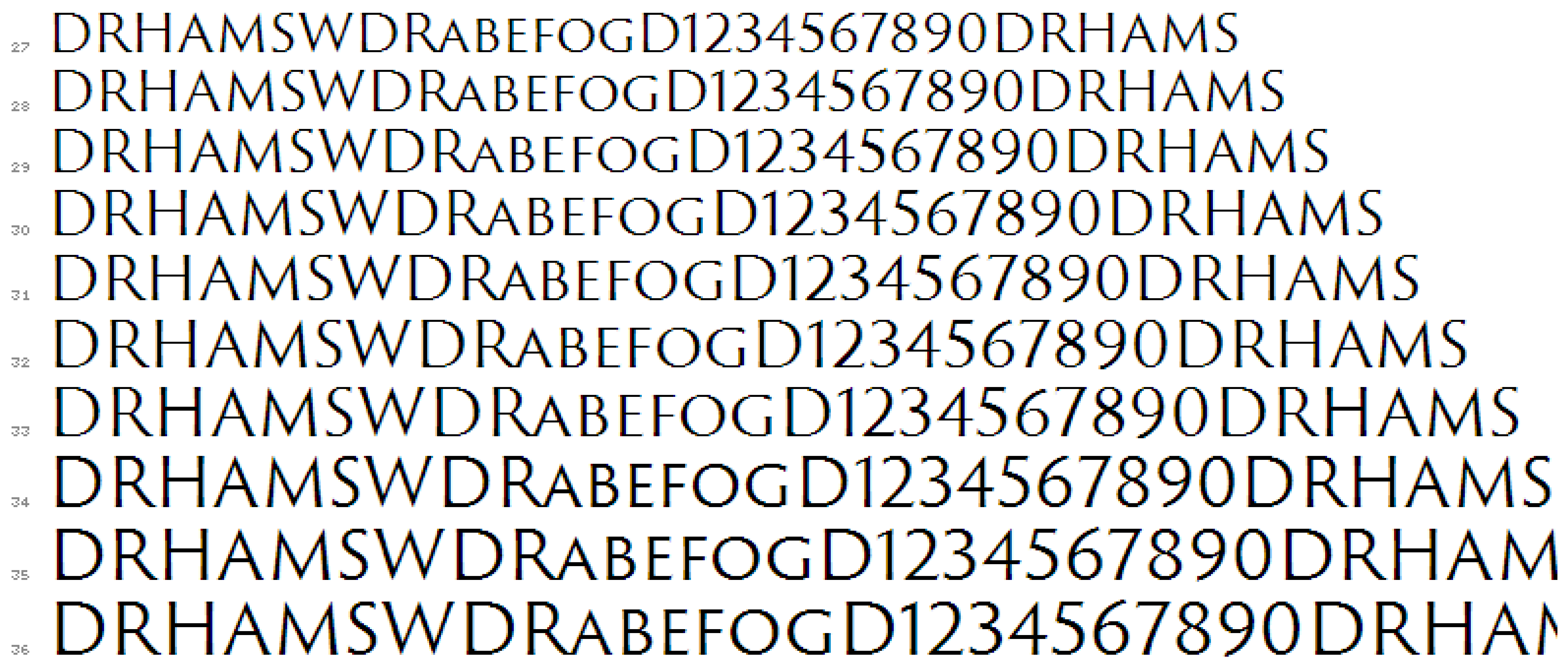


Name	Width	ppm1	ppm2	ppm3	ppm4	ppm5	ppm6
Y: 53	53	0	34	50	67	84	104
Y: 58	58	0	34	50	67	84	104
X: 83	83	0	21	30	42	54	67
X: 86	86	0	21	30	42	54	64

Y: 53 | 53 | 0 | 34 | 50 | 67 | 84 | 104


TrueType Preview
@RHAMSW@Rabefog@1234567890@RHAMSW
DRHAMSWDRABEFOGD1234567890DRHAMS
8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41
10 DRHAMSWDRABEFOGD1234567890DRHAMS
11 DRHAMSWDRABEFOGD1234567890DRHAMS
12 DRHAMSWDRABEFOGD1234567890DRHAMS
13 DRHAMSWDRABEFOGD1234567890DRHAMS
14 DRHAMSWDRABEFOGD1234567890DRHAMS
15 DRHAMSWDRABEFOGD1234567890DRHAMS
16 DRHAMSWDRABEFOGD1234567890DRHAMS
17 DRHAMSWDRABEFOGD1234567890DRHAMS
18 DRHAMSWDRABEFOGD1234567890DRHAMS
19 DRHAMSWDRABEFOGD1234567890DRHAMS
20 DRHAMSWDRABEFOGD1234567890DRHAMS
21 DRHAMSWDRABEFOGD1234567890DRHAMS
22 DRHAMSWDRABEFOGD1234567890DRHAMS
23 DRHAMSWDRABEFOGD1234567890DRHAMS
24 DRHAMSWDRABEFOGD1234567890DRHAMS
25 DRHAMSWDRABEFOGD1234567890DRHAMS
26 DRHAMSWDRABEFOGD1234567890DRHAMS
27 DRHAMSWDRABEFOGD1234567890DRHAMS
28 DRHAMSWDRABEFOGD1234567890DRHAMS
29 DRHAMSWDRABEFOGD1234567890DRHAMS
30 DRHAMSWDRABEFOGD1234567890DRHAMS
31 DRHAMSWDRABEFOGD1234567890DRHAMS
32 DRHAMSWDRABEFOGD1234567890DRHAMS
33 DRHAMSWDRABEFOGD1234567890DRHAMS
34 DRHAMSWDRABEFOGD1234567890DRHAMS
35 DRHAMSWDRABEFOGD1234567890DRHAM
36 DRHAMSWDRABEFOGD1234567890DRHAM

TrueType Hinting: PPM values



27 DRHAMSWDRABEFOGD1234567890DRHAMS

28 DRHAMSWDRABEFOGD1234567890DRHAMS

29 DRHAMSWDRABEFOGD1234567890DRHAMS

30 DRHAMSWDRABEFOGD1234567890DRHAMS

31 DRHAMSWDRABEFOGD1234567890DRHAMS

32 DRHAMSWDRABEFOGD1234567890DRHAMS

33 DRHAMSWDRABEFOGD1234567890DRHAMS

34 DRHAMSWDRABEFOGD1234567890DRHAMS

35 DRHAMSWDRABEFOGD1234567890DRHAM

36 DRHAMSWDRABEFOGD1234567890DRHA

Note the horizontal stem change from 1 to 2 pixels at 34 pt.

TrueType Hinting: Not magic!

Note:

All TrueType hinting examples show the approach of only hinting in direction of the y-axis, which means only taking care of fine-tuning the horizontal stems.

This is a common workflow today, which is supported by Windows' *ClearType* rasterizer. *ClearType* takes care of the vertical stems itself through sub-pixel rendering.

Of course, TrueType Hinting is also possible in the x-direction.