Sinhala Font Design Simplified: Analyzing Legacy Fonts to Identify the Minimum Glyph Resembling Shapes and Automating Unicode Compliant Font Generation with FontForge

H. A. P. De Silva

2024

Sinhala Font Design Simplified: Analyzing Legacy Fonts to Identify the Minimum Glyph Resembling Shapes and Automating Unicode Compliant Font Generation with FontForge

H. A. P. De Silva

2024

Sinhala Font Design Simplified: Analyzing Legacy Fonts to Identify the Minimum Glyph Resembling Shapes and Automating Unicode Compliant Font Generation with FontForge

A Thesis Submitted for the Degree of Master of Computer Science

H. A. P. De Silva
University Of Colombo School of Computing
2024

Declaration

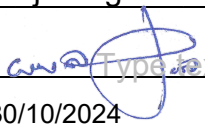
Name of the student: H. Achala Prabashwara De Silva
Registration number: 2018/MCS/011
Name of the Degree Program: Master of Computer Science
Project/Thesis title: Sinhala Font Design Simplified: Analyzing Legacy Fonts to Identify the Minimum Glyph Resembling Shapes and Automating Unicode Compliant Font Generation with FontForge

1. The project/thesis is my original work and has not been submitted previously for a degree at this or any other University/Institute. To the best of my knowledge, it does not contain any material published or written by another person, except as acknowledged in the text.
2. I understand what plagiarism is, the various types of plagiarism, how to avoid it, what my resources are, who can help me if I am unsure about a research or plagiarism issue, as well as what the consequences are at University of Colombo School of Computing (UCSC) for plagiarism.
3. I understand that ignorance is not an excuse for plagiarism and that I am responsible for clarifying, asking questions and utilizing all available resources in order to educate myself and prevent myself from plagiarizing.
4. I am also aware of the dangers of using online plagiarism checkers and sites that offer essays for sale. I understand that if I use these resources, I am solely responsible for the consequences of my actions.
5. I assure that any work I submit with my name on it will reflect my own ideas and effort. I will properly cite all material that is not my own.
6. I understand that there is no acceptable excuse for committing plagiarism and that doing so is a violation of the Student Code of Conduct.

Signature of the Student	**Date (DD/MM/YYYY)**
	30/10/2024

Certified by Supervisor(s)

This is to certify that this project/thesis is based on the work of the above-mentioned student under my/our supervision. The thesis has been prepared according to the format stipulated and is of an acceptable standard.

	Supervisor 1	**Supervisor 2**	**Supervisor 3**
Name	Viraj Welgama		
Signature	 Type text here		
Date	30/10/2024		

I would like to dedicate this thesis to my father, mother, sister and my loving wife Yashodha…

ACKNOWLEDGEMENTS

I would like to thank Prof. J. B. Disanayaka and Prof. Sandagomi Coperahewa, Department of Sinhala, University of Colombo. For their invaluable expertise in Sinhala language and their extensive contributions to the field through numerous publications over the years. And also, Dr. Sumanthri Samarawickrama, Faculty of Architecture, University of Moratuwa, for her invaluable research “The Anatomy and Historical Development of Sinhala Typefaces”, in my belief, research that filled a huge gap in Sinhala typography field.

Even though we will not be discussing in this thesis, it would be an oversight not to recognize the valuable contributions made by Dr. S. T. Nandasara, School of Computing, University of Colombo, and team in including Sinhala into Unicode standard. And also, the work of Information and Communication Technology Agency of Sri Lanka (ICTA) in incenting new font developers.

I would also like to extend my gratitude to my supervisor Mr. Viraj Welgama, Senior lecturer, University of Colombo. For guiding me throughout this journey. And to Aiden Catbagan who showed me that automating the font creation process is a reality and let me go through his work without hesitation.

Finally, I would like to extend my gratitude for all the font developers who spent days, weeks or months designing the Sinhala fonts we use today. Without their efforts the digital expansion of Sinhala would not be possible.

ABSTRACT

This thesis focusses on analyzing the ASCII based legacy fonts to identify common visual features of different glyphs in order to identify the minimum number of glyphs needed to be designed to create a Unicode compliant Sinhala font. Due to the lack of codepoints availability, most of the times, legacy Sinhala fonts consist only the glyphs which cannot be created by combining multiple glyphs. And since they are in common use among Sri Lankan people for almost 3 decades, this supports the hypothesis that the glyphs in these fonts can support most if not all Sinhala words in common use.

We identified that with 594 glyphs we can design a Unicode compliant Sinhala font that would facilitate the common Sinhala writing. In order to find the minimum glyph count needed to design these 594 glyphs we have suggested 6 main groups that the glyphs in a legacy font can be divided into. And we also suggested that glyphs in 2 of these groups can also be grouped into 8 sub-groups depending on their visual similarities.

Finally, we have identified that the minimum number of glyphs needed to be designed is 167 glyphs. And we also have suggested a way to reduce the number of glyphs to be designed to 81 by using glyphs from an open licensed font to generate glyphs for the codepoints in basic Latin character set.

With identifying the minimum number of glyphs to be designed we move to the second part of this research which is to generate a Unicode compliant font using the minimum glyphs designed by utilizing the python scripting capabilities of FontForge software. Here we have identified that there are 2 ways of doing this by either creating everything including lookup tables programmatically using python or using an existing font to create a new font by replacing its glyphs and other general information.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

In the realm of digital communication, fonts serve as the visual bedrock of written language, shaping the way we perceive and interact with information. Beyond mere aesthetics, fonts hold the power to evoke emotions, establish brand identities, and enhance the readability of textual content. Designing a Font is no simple task. An individual must possess both artistic inclination and technical literacy with a strong design sense and typography knowledge when it comes to becoming a font designer. In today's world there are thousands of fonts that represent many languages all around the globe.

Sinhala is one such language primarily used by the people of Sri Lanka. Being one of the two official languages in the country Sinhala is spoken by more than 17million people as their first language, specially by the Sinhalese people and more than 3 million people as their second language. Sinhala language consist of its own alphabet and a writing system which is a descendent of Brahmi script and also related to Kadamba script. Despite being a descendant of Brahmi script, the Sinhala letters have a curved shape rather than the angular shape of the letters in Brahmi script. ("Sinhala script," 2023; "The Sinhalese language," n.d.)

This research delves into the captivating world of Sinhala font design, seeking to untangle the complexities that have hindered its widespread proliferation. By analyzing legacy fonts, identifying common shapes, and proposing a streamlined approach to glyph design, this study aims to minimize the time it takes to create a Sinhala font. Thus, encouraging Sinhala font designers to create new fonts with minimal effort.

1.1 Motivation

The 1900s and early 2000s are often regarded as the golden era of Sinhala fonts. Many of the fonts widely used today were created during this time. A significant portion of these fonts was based on American Standard Code for Information Interchange (ASCII), and when Sinhala character set was included in Unicode standard, a new set of Unicode-based Sinhala fonts were developed. However, in the subsequent years, there has been a noticeable decline in the creation and release of new Sinhala fonts.

In 2009, the Information and Communication Technology Agency of Sri Lanka (ICTA) developed the Bhashitha font family, which consisted of 9 fonts. In 2009 and 2015 ICTA held training programs for font developing and several stylized Sinhala Unicode fonts were created

as a result. Despite these initiatives, the past decade has witnessed a relatively quiet period in terms of introducing new Sinhala fonts.

Around 2010, despite the introduction of the Sinhala Unicode standard and the availability of several Unicode-compliant fonts, ASCII-based legacy fonts continued to be widely used in various government offices and popular websites. This trend persisted for several reasons, mainly attributed to the inadequate support for the Sinhala language by operating systems and the limited usage of Sinhala on mobile devices. However, in recent years, these hindrances have been effectively addressed, marking a notable shift in the landscape.

The exponential surge in Sinhala's digital presence witnessed over the past two decades is evident through the proliferation of Sinhala-based websites and online content, as well as the growing engagement of the Sinhala-speaking community on social media platforms, seamlessly employing their native language. Moreover, the provision of translation support for Sinhala has further fueled this digital expansion. The introduction of new Sinhala fonts is crucial for the ongoing digital expansion. Relying on legacy fonts that have endured for several decades is no longer sustainable.

1.2 Statement of Problem

The current scarcity of Sinhala font developers is particularly pronounced in an era where the demand for digital content is more significant than ever before.

Creating a font entails a highly intricate process which might even take weeks or months, particularly when it comes to complex scripts such as Sinhala which contains complex character shapes where the same glyph can yield distinct shapes when combined with different vowel signs. Thus, the designer should take careful consideration of the individual character shapes and their interaction to create visually pleasing and well-proportioned fonts. Proficiency in vector designing software and font design tools is also indispensable for a font designer. An individual who composes all these abilities is a rare even in the computing community.

Due to these reasons, we see a very little number of font creators in Sri Lanka. Even though there are people with necessary skill set most of them tend not to be interested in font designing mainly due to the time and effort that can't be afford in the today's busy world.

Even for a legacy Sinhala font, which is based on ASCII, there needs to be around 200 - 230 different glyphs to be designed. For a Unicode compliant Sinhala font, the number of glyphs need to be designed greatly increases up to at least 500 glyphs. The process of font creation is

not bound to just glyph designing. it also consists of various other things such as giving further instructions on glyph substitution, contextual alternates, ligatures, and other typographic optimizations specific to Sinhala script that should be provided by the font designer. For an inexperienced individual, undertaking such a time-consuming commitment would become a vital factor diminishing their interest in font designing.

1.3 Research Aims and Objectives

This thesis centers on the analysis of glyphs within existing Sinhala fonts, aiming to identify visual resemblances among diverse glyphs. The objective is to leverage insights gained from this analysis to ascertain the capacity to generate glyphs based on the identified similarities in Sinhala script.

- Analyze the existing legacy fonts to identify the commonly used glyphs.
- Analyze existing legacy fonts to identify the commonly used shapes in the existing glyphs. And to determine the minimum number of glyphs that is needed to create a Unicode compliant Sinhala font.
- Propose a set of glyphs and design rules that is required to create a new font using the findings of the research.
- When a set of glyphs are provided that adhere to the above rules, generate a Sinhala Unicode compliant font.

1.4 Research Questions

- What is the minimum number of glyphs needed to create a Unicode compliant Sinhala font?
- Can we use the identified common parts of glyphs to construct new glyphs without designing them. If so, how many glyphs should be created to generate all the glyphs needed to be included in a Unicode compliant Sinhala font. And what are they?
- Is there any other way to reduce the minimum number of glyphs to be designed further?
- What is the possibility of using existing font to create a new font?

1.5 Scope

In this study, we aim to explore the inherent characteristics shared by various letters within the

Sinhala language. At first glance, it becomes evident that certain letters exhibit common features. To investigate this phenomenon, our research will utilize Sinhala legacy fonts to identify these shared attributes. The rationale behind employing legacy fonts lies in their widespread usage and extensive testing over time, enabling them to support a significant portion of Sinhala written words.

When developing a Unicode compliant Sinhala font, it is essential to include the entire basic Latin (ASCII) character set as well. Although our focus does not encompass the creation of all glyphs within the ASCII character set, we will include all numbers and commonly used punctuation marks. But we will not be focusing on the Sinhala numbers and the glyph 0D81 Chandra Bindu sign. We will not also be focusing on touching letters and conjunct letters which are not in common usage and already have a standard way of writing them.

1.6 Structure of The Thesis

This paper consists of 6 chapters. The introduction chapter will provide an overview of the thesis and its structure. The chapter 2 contains an overview of fonts, Sinhala language, researches done on the anatomy of Sinhala typeface, Sinhala legacy and Unicode fonts and finally about a font that was generated using python and FontForge's scripting tools.

The research can be divided into two primary phases. Firstly, we will analyze the glyphs of Sinhala legacy fonts and determine the minimum number of glyphs designs necessary to construct a Unicode compliant Sinhala font. Once these glyphs designs are identified and designed accordingly, the second phase involves generating the complete font. Chapter 3 of this paper will be focused on these 2 phases.

In the 4th chapter the implementation of the software purposed in the 3rd chapter will be discussed in a technological manner. In the 5th the fonts generated using the purposed software will be examined and discusses the evaluation criteria and methodologies used to assess the quality and effectiveness of the generated fonts.

The final chapter will summarize the key findings and contributions of the research and discuss potential areas for future research and development based on the findings.

CHAPTER 2
LITERATURE REVIEW

2.1 Fonts

In traditional typesetting the word “Font” was referred to a set of characters with a certain size, weight and style of a typeface (“Font,” 2024). A typeface can consist of one or more fonts shearing a common design. Each font has a matched set of pieces containing glyphs for each character, symbol or numeral. A glyph, which plays a major role in typography, can be defined as a bound mask of a letter in a script (“Glyph,” 2024). In Indic scripts a set of conjunct characters can form a single glyph. Although in traditional typesetting font and typeface are two distinct meanings, in modern typesetting the word font is also used as a synonym for typeface. In traditional typesetting it required to have separate fonts for different sizes. But in modern computer fonts the need of having separate fonts for each size is eliminated since modern computer fonts are scalable to any point size. But separate digital font file for each style is needed.

A computer font is a digital file that contains a set of glyphs. These fonts are created using a font editor like FontForge, FontLab, Fontographer etc. (“Font editor,” 2024). Screen fonts are special fonts that is designed only for computer screen and not for printing. Computer fonts can be grouped into 3 main categories.

- Bitmap fonts
- Vector fonts
- Stroke fonts

Digital fonts also contain information on font metrics which includes information on glyph substitution, kerning pairs, component creation data etc. Common fonts formats include True Type Fonts, Open Type fonts, Meta fonts and PostScript Type1 fonts.

2.1.1 Bitmap Fonts

Bitmap fonts, also known as raster fonts, stores each glyph as an array of pixels. They have a complete set of glyph images for each variant of the font like font size, weight etc. Despite the factor that large number of image sets are needed bitmap fonts have advantages like the rendering speed, ease of creating when compared with other categories, unscaled fonts giving same output on displays with same specification always etc. bitmap fonts are more suitable for

small-size or low-quality displays. These fonts also have disadvantages like poor quality when scaling and dramatic increasing of memory usage when font size changes etc. the limited hardware support in early computers forced the bitmap fonts to be used exclusively. But later with the hardware improvements outline or stroke fonts were used rather than bitmap fonts when scaling is desired. Yet bitmap fonts are still in use in systems like embedded systems where the simplicity and speed matter the most. Bitmap fonts looks best when displayed at their native pixel size. Some systems use different techniques to scale bitmap fonts such as nearest-neighbor resampling ("Nearest-neighbor interpolation," 2023), anti-aliasing ("Spatial anti-aliasing," 2024). These techniques work well for decreasing the font size but not for increasing the font size since the images are tend to have blur edges. A vector font can be created by determining the outline for a high-resolution bitmap font using a trace program. The same technique can be used when generating glyphs for the converted fonts in this project.

2.1.2 Vector Fonts

Vector fonts, also known as outline fonts, are a collection of vector images. They consist of lines and curves that defines the boundary of glyphs. Because of this unlike bitmap fonts vector fonts can be scaled to any size without causing any pixelation. But vector fonts require more processing power and often depend on the font, size and rendering software. Vector fonts use Bézier curves ("Bézier curve," 2024) which are hard to render accurately in a raster display. Rendering them can change the shape of the curve depending on the desired position and size. To reduce this, methods such as font hinting, a technique that use mathematical instructions to adjust the display of a vector font lining it up with a rasterized grid ("Font hinting," 2024), can be used. Like bitmap fonts used to create vector fonts, vector fonts can also be used for creating bitmap fonts if necessary. But this transformation is rather harder when converting curves. A heuristic algorithm is needed to determine and approximate the corresponding curves. This technique is also useful in this project since vector fonts are needed to be converted to bitmaps in order to identify the shape as a letter. Examples of outline fonts include: PostScript Type 1 and Type 3 fonts, TrueType fonts, OpenType fonts etc.

2.1.3 Stroke Fonts

In stroke baked fonts outline of a glyph is defined by the vertices of separate stroke paths. These paths can be defined as a topological skeleton like structure. Because of this the number of vertices needed to define a glyph reduces greatly and since the same stroke paths can be filled with different patterns these fonts can generate different sizes and weights or even different shapes. Editing stroke-based fonts are rather easy and less error prone when compared with bitmap or vector fonts. These fonts are heavily used on embedded devices specially for east Asian character sets since they claimed to save a lot of space.

2.1.4 True Type Fonts

Originally developed by Apple Inc and later licensed to Microsoft for free, True type fonts were introduced to replace Type 1 fonts which were very expensive. Instead of cubic Bézier curves like in Type 1 fonts true type fonts used quadratic Bézier curves. True type fonts are very popular because they allowed pixel level manipulation of the font giving font developers a great degree of control and freedom. However, pixel level control is no longer certain in a true type font due to rapidly varying rendering technologies in use today (“TrueType,” 2023). Currently all major operating systems supports this font format. Microsoft introduced smart font technology in 1994, which was known as True Type Open. it was later developed into Open Type Fonts.

2.1.5 Open Type Fonts

Built on True type fonts, Open type fonts retains the basic true type font structure while adding many complex data structures in order to prescribe the typographic behavior. Open type fonts are very important in non-western writing that have multiple characters or syllables forming a single glyph. Just as true type currently all major operating systems have rendering engines to support this font format. Uniscribe by Microsoft for windows operating systems, Pango for Linux are few examples. The instructions on how glyphs should be rendered are stored in open type fonts as Open type layout tables. these tables extend the functionality of fonts with either true type of CFF outlines. Rendering engines use these rules to conjunct multiple glyphs to create a single glyph or use an alternative glyph. Open type fonts include language information and other scripts that text processing applications use to adjust their behavior. Open type fonts are also created using font creating tools. And same or different software can be used to create the rules needed for the rendering process of the fonts.

2.1.6 Open Type Font rules

The rendering information are stored in Open type tables some commonly used tables for non-western writing are as follows.

- GSUB: contains glyph substitution related information. This information is used in single, one-to-many, multiple or contextual substitutions
- GPOS: contains X, Y positioning information of glyphs to handle adjustments on single or paired glyphs, cursive or mark attachment and contextual positioning of glyphs.
- BASE: contains baseline offsets information on a script-by-script basis.
- JSTF: contains information about justification with whitespace and kashida adjustments.
- GDEF: contains information about all the glyphs in the font.
- CMAP: character to glyph mapping
- Head: font header
- Post: PostScript information
- Name: naming Table

The Open type layout model is organized around glyphs, scripts, language systems and features. A Script consist of a set of related characters that used in one or several languages. A font can be a single or multiple scripts. Scripts are identified by unique 4-byte tags in Open type fonts. Scripts can then be divided into language systems. For example, both English and French languages use the Latin script. Providing the information that is tailored to the language system, script or both is up to the font developer. The basic functionality of a font is defined by features. These features are often defined by the language itself. Whenever language features are not available default features will be applied. Features of open type fonts are implemented using lookup tables.

2.2 Sinhala

Sinhala is an Indo-Aryan language primarily used by the people of Sri Lanka. Being one of the two official languages in the country Sinhala is spoken by more than 17million people as their first language, specially by the Sinhalese people and more than 3million people as their second language. The Sinhala language is considered to originated around 5th century with the colonists, including prince Vijaya who is said to be the first Sinhalese king, from northern India. At that time the native tribes in Sri Lanka, known as the Hela tribes, were using a language called Elu ("Elu," 2024). The Indian colonists and native tribes were mixed relatively well and their languages Prakrit ("Prakrit," 2024) and Elu were also combined and formed the Proto-Sinhala language over the following hundred years. Proto-Sinhala was used in the period 3rd century to 7th century. (Aliyar, 2011; "Sinhala language," 2024)

The Medieval Sinhala period was stemmed from the 7th century to the 12th century. Up to this point the language had gone through a lot of changes over the time. After 12th century begins the modern Sinhala period and it is the language that is still in use to this date and has stayed the same without going through drastic changes for most parts. which means the modern users of Sinhala language can easily understand texts that was written as far as the 12th century. ("Sinhala language," 2024)

Since being an island nation Sinhala was isolated from the other various Indo-Aryan languages that existed in India and the language was developed independently of those languages. Yet Sinhala language was greatly influenced by Pali language ("Pali," 2022), which was the sacred language of Buddhists, and Sanskrit ("Sanskrit," 2022) for some degree. It was also influenced and adopted various words and grammatical structures from Dravidian Languages like Tamil language. With the colonization Sinhala language had also picked up some words from Portuguese, Dutch and English languages. ("Sinhalese language | Sri Lanka, Indo-Aryan, Pali | Britannica," n.d.)

Sinhala language is special for many reasons and one of them is the Diglossia which is not seen very often in vernaculars. This means the language has two different versions, one version for everyday use, mainly for speaking and the other one is for formal occasions, mainly for writing. ("Sinhala language," 2024)

2.2.1 Sinhala Script

Sinhala language consist of its own alphabet and a writing system which is a descendent of Brahmi script (“Brahmi script,” 2024) and also related to Kadamba script (“Kadamba script,” 2024). Despite being a descendant of Brahmi script, the Sinhala letters have a curved shape rather than the angular shape of the letters in Brahmi script. The reason for this is said to be the usage of dried palm leaves for writing in ancient Sri Lanka for centuries. In order to write on palm leaves the letters were carved in to the palm leaves using metal tools. With the letters with straight or angular form there was a greater risk of the leaves getting teared apart. With curved letters however the risk of tearing the leaves were minimized and they were easier to curve thus the letters in Sinhala script were developed to the curved letters seen in present day. (“Sinhala script,” 2023; “The Sinhalese language,” n.d.)

2.2.2 Structure

Like a lot of Indian scripts, Sinhala writing system is also a syllabic alphabet which is written from left to right. When constructing words, a consonant is used as the basic unit. Each consonant has an inherent vowel, which is a vowel sound used with a basic consonant. In Sinhala language the vowel අ /a/ is used as the inherent vowel. In order to represent deferent phonemes, the inherent vowel can be changed with other vowel strokes or signs. That can be used before, after, below or above the base consonant. Whenever a vowel is used as the starting letter of a word it will be written in its original form. The basic form of the letter n is න since the inherent vowel is /a/ this can be written as “na”. To create a pure consonant that does not have a vowel following the consonant, a special marker called “hal kirima” is used. It suppresses the inherent vowel.

න = na | න + ් = න් = n

2.2.3 Signs

Signs are called pili / pillam in Sinhala language. Using these around a consonant can create different phonemes.

Table 2.1: Signs in Sinhala language

Pilla	*Name*	*Formation*	*Compound Form*	*Special Cases*
්	Hal kirīma	ක්	ක්	The common usage
		ම	ම්	Whenever the letter is ending at

				top left corner
◌	-	ක් + අ	ක	
◌ා	Ælapilla	ක් + ආ	කා	
◌ැ	Ædaya	ක් + ඇ	ක	
		ර + ඇ	රැ	Since the diacritic is used for රැ idiosyncratic form is used to represent this
◌ෑ	Diga ædaya	ක් + ඈ	කෑ	
		ර + ඈ	රෑ	Since the diacritic is used for රෑ idiosyncratic form is used to represent this
◌ි	Is-pilla	ක් + ඉ	කි	
◌ී	Diga is-pilla	ක් + ඊ	කී	
◌ු	Pā-pilla	ක් + උ	යු	The common usage
			කු	When the letter is ending at the lower right corner. But not applicable for letters න and
			රු	When vowel /u/ is used with letter ර
			ළු	When vowel /u/ is used with letter ළ
◌ූ	Diga pā-pilla	ක් + ඌ	යූ	The common usage
			කූ	When the letter is ending at the lower right corner. But not applicable for letters න and
			රූ	When vowel /ū/ is used with letter ර
			ළූ	When vowel /ū/ is used with letter ළ
◌a	Gæṭa sahita ælapilla	ක් + ර් + උ	කa	
◌aa	Gæṭa sahita ælapili deka	ක් + ර් + ඌ	කaa	

්ෟ	Gayanukitta		කෟ	
ෙ	Kombuva	ක් + ඒ	කෙ	
ේ	Kombuva saha halkirīma	ක් + ඒ	කේ	
ෛ	Kombu deka	ක් + ඓ	කෛ	
ො	Kombuva saha ælapilla	ක් + ඔ	කො	
ෝ	Kombuva saha halælapilla	ක් + ඕ	කෝ	
ෞ	Kombuva saha gayanukitta	ක් + ඖ	කෞ	

There are also several non-vocalic signs in Sinhala language. They are ◌ං which is called anusvara / binduwa and ◌ඃ which is called visarga.

There are some symbols that can be used instead of several consonants.

Table 2.2: Symbols to be used instead of consonants

yanssaya	Used instead of the “ය” after a pure consonant	විද්‌යාව	විද්‍යාව
rakāransaya	Used instead of the “ර” after a pure consonant	මිත්‌ර	මිත්‍ර
rēpaya	Used instead of the “ර්” before a consonant	ධර්ම	ධර්‍ම

Unlike in English language signs can be positioned around a character and they can be categorized as left, right, upper and lower modifiers.

- Left modifiers: ෙ ෛ
- Right modifiers: ා ැ ෑ ෘ ෲ ෟ ං ඃ ෳ
- Upper modifiers: ි ී ු ූ
- Lower modifiers: ෘ ෲ ෟ

2.2.4 Sinhala Alphabet

Due to the changes in the language over time there can be seen several alphabets in Sinhala language but within all these alphabets the core of the language remains unchanged. Disanayaka and Coperahewa (Disanayaka, n.d.; Coperahewa, n.d.) both have listed 7 such alphabets in their books.

- Sidath Sangara hōdiya
- Amiśra Sinhala hōdiya
- Pansal hōdiya
- Wadan Kavi hōdiya
- Miśra Sinhala hōdiya
- National Institute of Education alphabet
- International Computing Sinhala alphabet / Sinhala Unicode alphabet

Disanayaka has included an additional alphabet named Modern Sinhala alphabet which removed unused characters in the modern Sinhala language and including several much-needed letters for writing English words.

2.2.5 Sinhala Unicode

The first version of Unicode, Unicode 1.0, was published in 1991, it did not contain the Sinhala character set. Sri Lanka submitted a proposal for the Sinhala character code at the Unicode working group meeting in Crete, Greece in 1997. There were other proposals from UK, Ireland and USA. Finally, it was accepted over the other competing proposals with slight modifications and ratified at the meeting of the working group held at Seattle, USA in 1998. The Sinhala code chart was included with Unicode 3.0 which was released in September 1999. It contained 80 characters for Sinhala script. The SLS 1134 was also revised accordingly in 2001. (Nandasara et al., 2003)

There were several slight modifications for Sinhala in the following versions.

- Unicode 7.0 – Sinhala Archaic Numbers were added.
- Unicode 13.0 – Including character used in Sinhala to write Sanskrit for additional support for lesser-used languages and scholarly work.

The allocated range for Sinhala in Unicode code table is 0D80-0DFF.

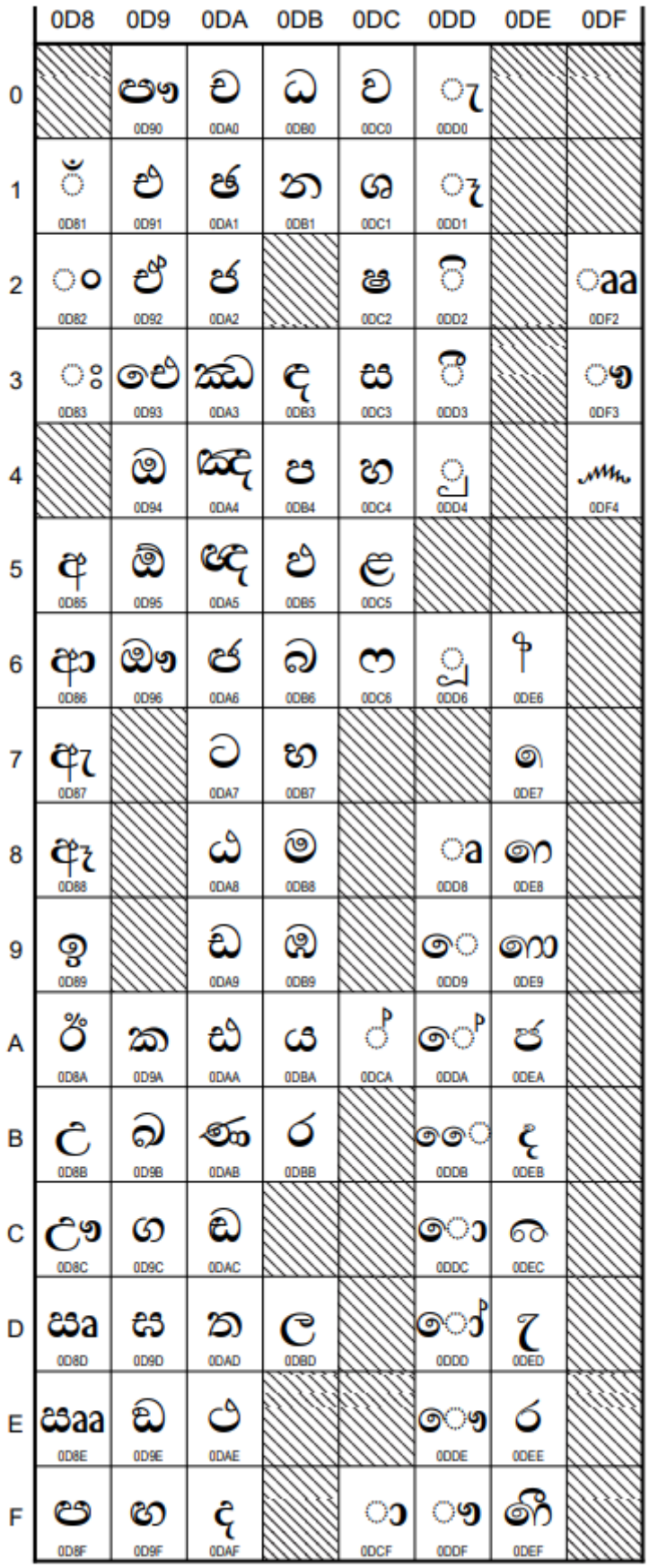


Figure 2.1: Sinhala Unicode character encoding
Source: (“The Unicode Standard, Version 15.1,” n.d.)

The focus of this thesis is on this Sinhala alphabet accepted by the International Standards Organization (ISO). Out of the 80 characters. This alphabet consists of 61 letters. Out of these 18 were vowels and the other 43 were consonants. The specialty of this alphabet is that the consonant ඈ being included which was not available with any other Sinhala alphabet.

Unicode standard only addresses the encoding and semantics of text. Any other action is addressed by the text processing software. And the text rendering is handled by the hardware or software rendering engine. Unicode standard does not define how glyphs are rendered it only defines how characters are interpreted.

Hence the 18 signs with another special symbol “kundaliya” are included thus making 80 characters altogether.

2.3 Anatomy of Sinhala Typeface

When classifying Sinhala letters Disanayaka has shown that Sinhala letters can be classified under several criteria (Disanayaka, 2006). One such criteria is the starting and ending components of letters. He classifies them into 2 main groups.

- Starting letter components (aarambaka aksharanga)
- Ending letter components (samapthika aksharanga)

The starting letter components are then divided to 10 variations.

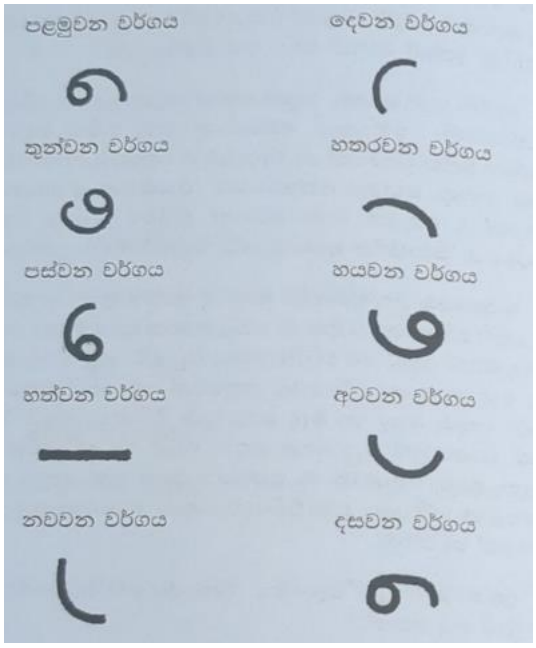


Figure 2.2: Starting letter components
Source: (Disanayaka, 2006)

The ending letter components are also divided to 10 variations.

Figure 2.3: Ending letter components
Source: (Disanayaka, 2006)

Another way of classifying Sinhala letters suggested by both Ariya Wikrama and Disanayaka is according to the height of the letters. Despite the usage of different names, the idea seems to be the same. Which, classifies letters into 3 main groups

- Base letters
- Ascending letters
- Descending letters

Disanayaka's classification was based on 4 parallel rows. These rows were called Top line (udu pela), Intermediate line (athuru pela), Middle line (meda pela) and Bottom line (yati pela) as shown in figure 2.4.

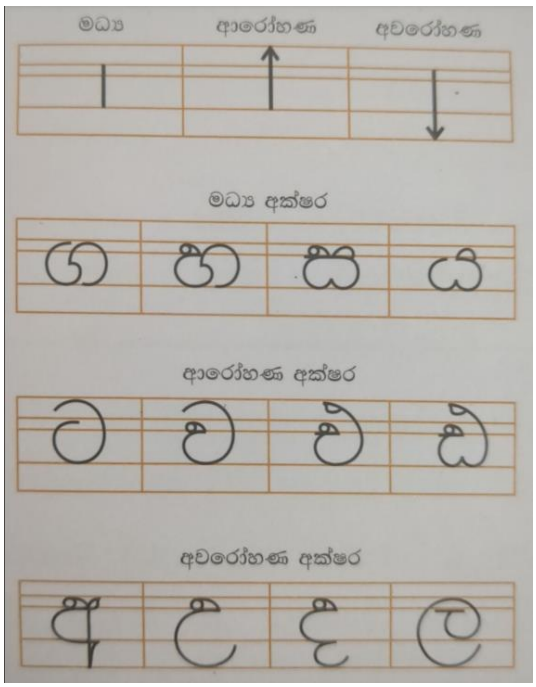


Figure 2.4: Letter classifications
Source: (Coperahewa, n.d.)

2.3.1 Five reference Lines

With the aim of improving handwriting of school children and as a guide for school teachers National education commission published a guide book named “A Guide to Write Sinhala Letters” (Sinhala Akuru liveema sandaha margopadesha) in 2005. It proposed that it is suitable to construct the Sinhala letters within five parallel lines. This was recommended for the 1st grade students since 2016. (Coperahewa, n.d.)

Due to the nature of the Sinhala letters these 5 lines are not of the same height. All base letters are written in between lines 2 and 4. The ascending letters are written between line 1 and 4 while the descending letters are written in between line 2 and 5. If we take a base letter The main part, which is usually the part that take the most proportion of a letter is written in between lines 3 and 4. While the part containing the eye is written in between line 2 and 3. If we divide the total distance between line 1 and line 5 in to 8 equal portions, then, the distance between each line can be considered as 2 portions between line 1-2, 1 portion between line 2-3, 3 portions between line 3-4, 2 portions between line 4-5.

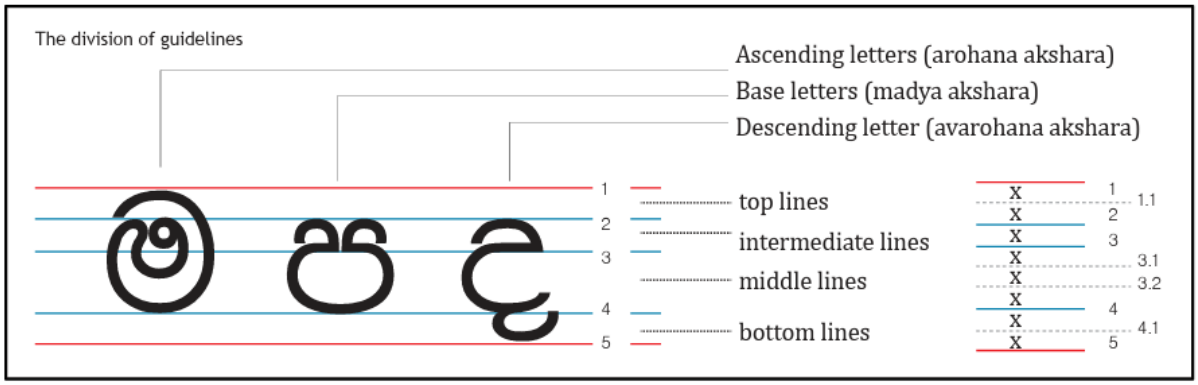


Figure 2.5: The five letter guidelines of the NIE publication
Source: (Samarawickrama, 2017)

There are several other guidelines that were suggested such as six reference lines, four reference lines and Circular Grid. Samarawickrama discuss about these guidelines in detail in her research paper "The Anatomy and Historical Development of Sinhala Typefaces". (Samarawickrama, 2017)

2.3.2 The Anatomy and Historical Development of Sinhala Typefaces

In this paper Samarawickrama done an analysis of the visual appearance of Sinhala letters and group them into 2 tables. The first table is based on the construction of the letter which is shown in figure 2.6 and the second table is based on the circular nature of Sinhala letters as ascender, descender or base characters. This is shown in figure 2.7.

Construction	Horizontal	Vertical
Two-storied	ඍ ත ග ඹ ඟ ඩ ණ ත න භ ය ශ ස හ ෆ	ඉ ඬ ද ඳ
Open	ෆ	
Wide	ඩ ඬ ණ	ම බ ඹ ධ ඩ ඪ ඹ ධ බ ඪ
Medium Wide	ඍ ඦ ක ග ඹ ඟ ත න භ ය ශ ස හ ෆ	එ ම ව ව ච ට ඨ අ ඉ
None	—	ඊ ර ද ඳ

Figure 2.6: Visual similarities on the structural formation of Sinhala letter by Samarawickrama

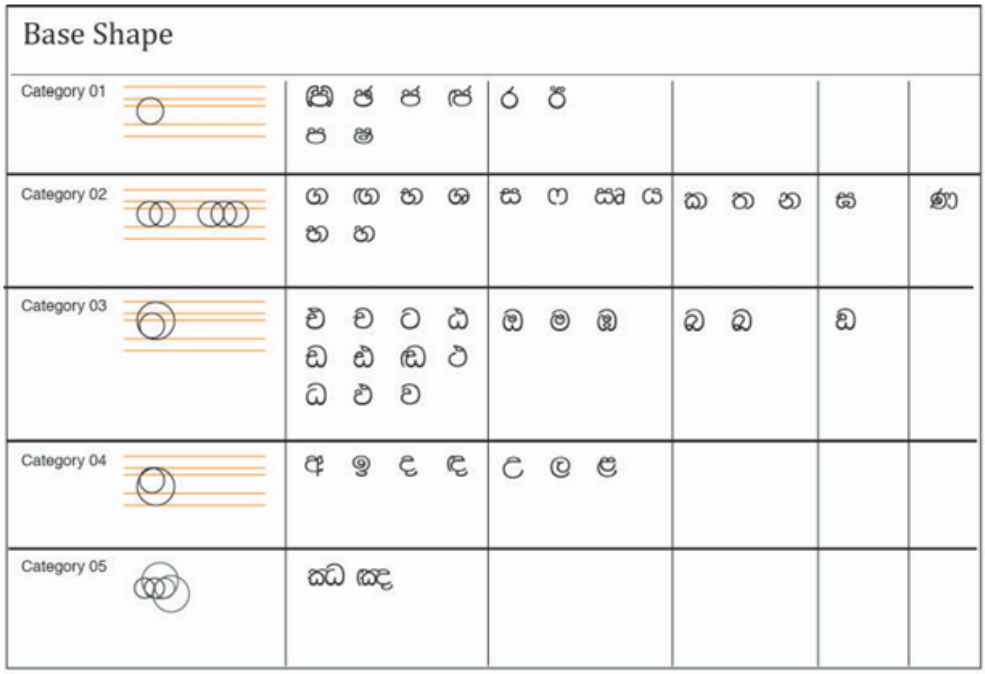


Figure 2.7: The circular nature of the Sinhala letters as: ascender, descender and base characters by Samarawickrama

Samarawickrama's research addresses the scarcity of studies and observations concerning the morphological characteristics of Sinhala typefaces. It encompasses a broad scope, ranging from the identification and categorization and terming of visual attributes of Sinhala letters to the proposal of a grid for developing novel Sinhala typeface designs. Consequently, it emerges as a seminal contribution to the field of Sinhala typography research.

Figure 2.8: Names given for distinct visual properties identified by Samarawickrama

2.4 Sinhala Font Levels

Sinhala fonts come in three standardized levels, each offering varying degrees of support. Level 1 fonts provide fundamental Sinhala characters, while higher levels incorporate more advanced features and functionalities.

- Level 1: These fonts typically cover basic Sinhala characters and are suitable for general text purposes. Level 1 fonts are designed to support the fundamental Sinhala character set without complex ligatures or special characters.
- Level 2: these fonts offer more advanced features compared to Level 1 fonts. They may include additional ligatures, conjunct characters, and diacritics, providing better support for complex writing styles and typographical requirements.
- Level 3: These are the most comprehensive and advanced fonts. They cover the full range of Sinhala characters, including rare or archaic glyphs, stylistic alternates, and specialized symbols. Level 3 fonts are typically used for specialized purposes such as typography, publishing, or academic research.

Each level builds upon the previous one, offering greater support for different writing styles, linguistic nuances, and typographical requirements. The choice of font level depends on the specific needs of the project or application. In this research we will be focusing on creating a font that meets Level 2 requirements. But we will include all the special letters and letters and all combinations of strokes with conjuncts which supposed to be included in Level 3 fonts. but we will not include the touching letters and conjunct letters in this research.

2.5 Creating Sinhala Unicode Fonts

The ICTA has undertaken significant efforts for the betterment of Sinhala Unicode. Recognizing the shortage of local font developers, ICTA proactively addressed this issue by disseminating font rules for Bhashitha and Sri Tamil to local font developers, facilitating the font creation process. Additionally, various font development programs were conducted to enhance awareness among developers, focusing on both technical aspects, such as font rules, and design principles.

A notable initiative in this regard was the font development camp organized in collaboration with the University of Colombo School of Computing (UCSC) in 2009. Subsequently, a comprehensive guide titled “Guide to Create Sinhala and Tamil Unicode Fonts” was published in 2010, compiling the proceedings of the training program. This publication aimed to support

local font developers in crafting high-quality, standards-based Sinhala fonts. (Wijayawardhana and Goonetilleke, 2010)

Creation of a Unicode compliant font involves few steps,

- Glyph Design: The glyphs that needs to be included in the font should be designed digitally using a software like Adobe illustrator, Glyphs, Corel Draw.
- Glyph Encoding: Using a software like FontForge, Font Lab the font developer should map the designed glyphs with the relevant code points.
- Implementing Substitution Rules: when displaying a glyph, it may have special behaviors that are established in the language. And sometimes the positions of glyphs might need to be slightly be changed with the glyph around it. Ancher points are used here. The rules relevant to these characteristics should be defined. The substitution rules of several Sinhala Unicode will be discussed later in this chapter.
- Testing: After creating the font, it needs to be thoroughly tested to ensure that all glyphs display correctly and that there are no issues with spacing, kerning, or rendering. Testing should be done on different operating systems, devices, and applications to ensure compatibility.
- Refinement: Based on testing feedback, font developer may need to refine the font by adjusting glyphs, kerning pairs, or hinting instructions to improve its appearance and readability.
- Distribution: Once the font is finalized, it can be distributed to users.

Samarawickrama states a concerning trend of existing typefaces being derived from one another, lacking originality. This was also proven when gathering fonts for our research, out of 1777 examined founts which were downloaded online, 1215 fonts were found to be duplicates when compared by meta data, while the remaining 562 fonts showed significant similarities or minor modifications, with some even containing unused glyphs from other fonts. Samarawickrama explains that this practice of replicating existing fonts was primarily motivated by the demand for Sinhala type within a rapidly evolving technical landscape, further facilitated by the absence of copyright laws specifically applicable to type designs. This practice seems to be developed due to the following reasons,

- Good reference point: A Sinhala font can contain a large number of glyphs with at least a minimum of 200 glyphs. So being able to visually see and easily identify which glyphs needs to be replaced makes the process easy for the developer.
- No need to redefine rules: Sinhala Unicode fonts have a large number of substitution rules if it were to define them manually it would take an extreme amount of time. So, it makes sense to use existing rules and abiding to them rather than defining them manually.
- Saves time: this practice reduces a significant amount of time that a font developer spends in steps 2 and 3 when developing a font.

2.6 Analysis of Sinhala Legacy Fonts

The legacy Sinhala fonts are based on the extended ASCII which is the 8-bit variant of ASCII. The 7-bit ASCII was not enough to cater the needs of Sinhala script, much like other Asian scripts, because it only supported 128 code points. However, the 256 codepoints in extended ASCII were manageable in crating Sinhala fonts for day-to-day usage.

When we examine legacy fonts, we come across 2 font types, “FM fonts” and “DL fonts”. They refer to different font technologies used for rendering Sinhala script on digital platforms

FM Fonts (Fixed Width Fonts): These fonts have a fixed width for each character, meaning each character occupies the same amount of horizontal space regardless of its actual width. FM fonts are commonly used in text files where aligning characters vertically is important. However, they are less commonly used for general Sinhala text rendering due to their less aesthetic appearance.

DL Fonts (Dynamic Layout Fonts): These, on the other hand, employ dynamic layout techniques to adjust the width of characters based on their shape and context within a word. This results in a more visually appealing and natural rendering of Sinhala text, as characters can adjust their width to fit together more harmoniously. DL fonts are preferred for general text rendering, especially in digital publishing and web environments where aesthetic appeal is important.

When examined fonts from both these types the and several observations were made. For fonts of FM type, all these fonts seem to have 221 code points without considering the empty glyphs. these fonts usually have more than 20 empty glyphs. Apart from the empty glyphs these fonts consist of 6 glyphs for roman letters 7 glyphs containing symbols and the rest containing either

letters numbers, punctuation marks or pillam. The similarity between all these fonts might be due to the fact that the same person, Mr. Pushpananda Ekanayaka who is one of the most influential people in Sinhala typography field, designed all these fonts.

For Fonts of DL Type, the number of glyphs differ in each font but the codepoints and corresponding glyph are the same, even though the font developer is different, in all these fonts. If we compare a FM and DL font, in some cases, the glyph for the same code point is different between a FM and a DL font. Which is denoted in figure 2.9 comparing a set of glyphs from FM-Bindumathi and DL-Araliya fonts.

- Fonts analyzed for FM Type: FM-Bindumathi, FM-Abhaya, FM-Derana, FM-Malithi
- Fonts analyzed for DL Type: DL-Araliya, DL-Champika, DL-Divani, DL-Manel

Figure 2.9: Different glyphs for the same codepoint in FM and DL fonts

These fonts don't contain any substitute rules or anchor points. both GSUB and GPOS tables are empty. This was a common characteristic of the legacy Sinhala fonts. When typing with these fonts a special input method editor (IME) was needed. Keyrep, Keyman, Sinhala Tamil IME were famous software which facilitate this.

But this on the other hand is proof that the glyphs on these fonts were enough for regular Sinhala usage. And supports the decision to base the research on analyzing Sinhala legacy fonts.

2.7 Analysis of Sinhala Unicode Fonts

In order to analyze Sinhala Unicode fonts, we would take Bhashitha1, BhashitaComplex developed by Information & Communication of Technology Agency of Sri Lanka (ICTA) and IskolaPota developed by Microsoft.

Table 2.3: Comparison of Sinhala Unicode fonts

	Bhashitha1	*BhashitaComplex*	*IskolaPota*
Glyph Count	277	1117	828
File Size	250 KB (256,228 bytes)	1.09 MB (1,150,128 bytes)	525 KB (538,564 bytes)
Anchor Points	Yes	Yes (but not on main glyphs)	Yes (only on 25CC (Dotted Circle))
GSUB	70 Lookups	70 Lookups	9 Lookups
GPOS	35 Anchor Classes	5 Anchor Classes	3 Anchor Classes

Bhashitha1 is the font with the lowest number of glyph count but it utilizes the existing glyphs to successfully create new glyphs thanks to the large number of anchor-points it has. An anchor point is a reference point used to define the position of certain elements within a glyph. Here, they are used to align the pillam on each letter without disturbing the visual appearance.

However, in both BhashitaComplex and IskolaPota the usage of anchor points is reduced. This was done by using separate glyphs for each variant of a letter with pillam. Thus, increasing the glyph count in the font but reducing the number of anchor classes needed. In iskola pota the only anchor class used is for the dotted circle which will be displayed if a “pillama” was typed without a base letter.

Instead, the use of glyph substitution rules was used. whenever the key combination is given the multiple glyphs will be replaced by a single glyph which is visually more accurate. Even though IskolaPota has 9 lookup tables in GSUB the sub tables for each lookup contains more than the number of rules in Bhashitha fonts.

2.8 Font Generation Using Python and FontForge

In 2021, Aiden Catbagan, a youtuber and a developer came across Kakuji (角字) which is a style of characters used for creating seals which became popular during the Edo era in Japan. Aiden resolved to create a proper Kakuji font. Despite lacking prior experience in font creation, Aiden recognized that manually crafting the glyphs, which consist of blocks, would be extremely time-consuming. Consequently, he turned to Font Forge's Python scripting feature for assistance.

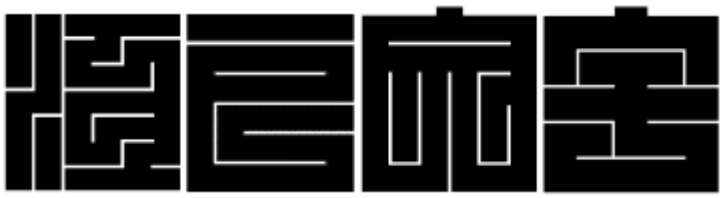


Figure 2.10: Kakuji characters
Source: ("Edomoji," 2023)

Aiden's approach to crafting the Kakuji font was straightforward. He intended to design the glyphs in Photoshop, selecting a standard set of 200 glyphs to be created and saved in PNG format. These images would then undergo analysis using Python, wherein each pixel would be examined to distinguish between the black and white sections, ultimately forming the basis for generating and incorporating glyphs into the font file through FontForge's Python scripting capabilities. The resulting font file would be exportable in TrueType format, suitable for regular usage as a conventional font.

Aiden's vision for the Kakuji font extended beyond its initial scope. Recognizing its intended purpose for seals, he expanded its repertoire to encompass additional glyphs, such as Japanese family names and diverse characters like zodiac signs. Furthermore, he introduced another font variant, New Kakuji Bold, which offered an inverse style to the original font. Aiden's comprehensive approach also incorporated English letters and common punctuation marks. This extensive collection culminated in the creation of a commercially available font package known as New Kakuji.

Figure 2.11: New Kakuji
Source: (“New Kakuji Font | Webfont & Desktop | MyFonts,” n.d.)

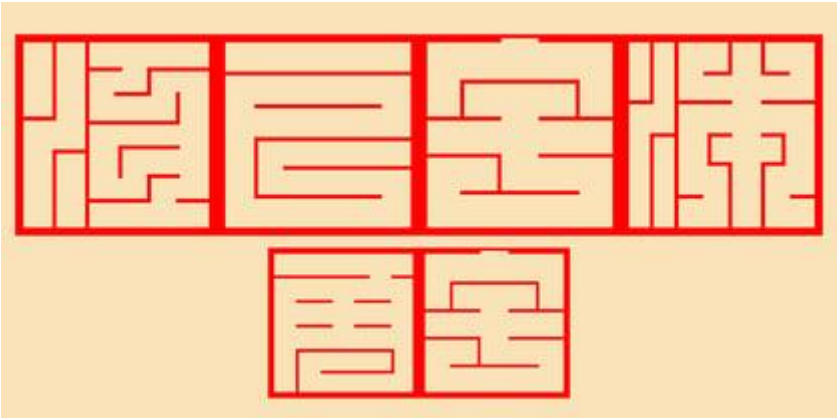


Figure 2.12: New Kakuji Bold
Source: (“New Kakuji Font | Webfont & Desktop | MyFonts,” n.d.)

Aiden’s comments on his work were, “I’m not a designer. I’m a programmer who also owns Photoshop. Without FontForge’s scripting tools I would not have been able to design this font at all.” This statement underscores the potency of FontForge’s scripting tools in enabling font creation, particularly for individuals like Aiden who possess programming skills rather than traditional design expertise. It emphasizes that with access to the necessary glyphs, FontForge’s scripting capabilities can effectively handle the font creation process.

2.9 Conclusion

The development of the New Kakuji Font underscores the potential for streamlining the font generation process when approached with careful design considerations. By leveraging FontForge's Python scripting tools to simplify steps 2 and 3 of the font creation process, our focus can shift towards optimizing glyph creation. This involves utilizing existing glyphs to generate additional ones, thereby reducing the number of glyphs that require manual design. While we won't be implementing Samarawickrama's proposed grid, the insights gleaned from her research, along with contributions from other scholars in the field, will be used extensively in this research.

CHAPTER 3
METHODOLOGY

In this research, an analysis of glyphs in various legacy fonts was conducted. A dataset comprising 1,777 Sinhala fonts sourced from the internet was initially acquired. Subsequently, a pruning process was initiated using a Python script in conjunction with FontForge, aimed at identifying and eliminating duplicate fonts and corrupted files. Following this process, 562 fonts remained in the dataset. The glyphs from these fonts were then extracted in Portable Network Graphics (png) format. After pruning, any non-Sinhala fonts were excluded, resulting in a final dataset of 475 valid Sinhala legacy fonts.

The next step involved grouping the extracted glyphs, utilizing relevant Unicode codepoint names for future ease of reference. A total of 588 glyphs were identified, including punctuation marks and numbers, while 204 Latin glyphs were deemed outside the scope of this research and subsequently excluded.

Table 3.1: Glyph naming

ක	ක	0D9A	0D9A
ක්	ක + ්	0D9A + 0DCA	0D9A0DCA
කි	ක + ි	0D9A + 0DD2	0D9A0DD2
කී	ක + ී	0D9A + 0DD3	0D9A0DD3
කු	ක + ු	0D9A + 0DD4	0D9A0DD4
කූ	ක + ූ	0D9A + 0DD6	0D9A0DD6
ක්‍රි	ක + ් + ර + ි	0D9A + 0DD2 + rakaranshaya	0D9A0DD2.rakar
ක්‍රී	ක + ් + ර + ී	0D9A + 0DD3 + rakaranshaya	0D9A0DD3.rakar

The third step involved identifying which glyphs were available in the dataset of Sinhala fonts and determining the percentage of fonts containing each glyph. Glyphs that were present in over 75% of the fonts were listed along with their corresponding Unicode codepoints. Additionally, glyphs that exist in the Sinhala Unicode standard, regardless of their availability in the dataset, were included as shown in Table 3.2.

Table 3.2: Glyph availability in legacy fonts

.0DD4	95.79831932773
.0DD6	96.00840336134
.notdef	99.36974789915
0020	100.4201680672
0025	96.00840336134
0028	96.84873949579
0029	99.36974789915
002C	100.0
002D	97.68907563025
002E	99.78991596638
0030	96.84873949579
0031	96.63865546218
0032	100.0
0033	100.0
0034	100.0
0035	100.0
0036	100.0
0037	100.0
0038	100.0
0039	100.0
003A	95.16806722689
003F	99.36974789915
0D85	100.0
0D89	100.0
0D8A	99.36974789915
0D8B	100.0
0D91	98.94957983193
0D94	100.0
0D95	96.00840336134
0D9A	97.26890756302
0D9B	99.36974789915
0D9C	99.36974789915
0D9D	97.89915966386
0D9F	95.79831932773
0DA0	97.26890756302
0DA1	98.94957983193
0DA2	99.57983193277
0DA4	97.89915966386
0DA5	96.84873949579
0DA7	97.89915966386
0DA70DCA	96.21848739495
0DA8	99.15966386554
0DA9	97.68907563025
0DAA	97.26890756302
0DAB	100.0
0DAC	96.21848739495
0DAD	99.36974789915
0DAE	99.57983193277
0DAF	98.94957983193
0DB0	98.73949579831
0DB00DD2	96.21848739495
0DB1	97.68907563025
0DB3	96.00840336134
0DB4	97.68907563025
0DB5	98.73949579831
0DB6	99.36974789915
0DB60DCA	95.58823529411
0DB7	98.10924369747
0DB8	100.0
0DB9	97.47899159663
0DB90DD2	95.79831932773
0DBA	100.0
0DBB	100.0
0DBB0DD1	95.58823529411
0DBD	100.0
0DBD0DD4	96.84873949579

0DBD0DD6	96.63865546218
0DC0	99.78991596638
0DC00DCA	96.00840336134
0DC1	99.57983193277
0DC2	97.26890756302
0DC3	99.15966386554
0DC4	99.57983193277
0DC5	98.31932773109
0DC50DD4	97.05882352941
0DC6	99.57983193277
0DCA	95.58823529411
0DCF	97.89915966386
0DD0	100.0
0DD1	98.52941176470
0DD2	95.58823529411
0DD3	96.00840336134
0DD4	96.00840336134
0DD6	95.16806722689
0DD8	99.57983193277
0DD9	100.0
0DDF	100.0
yanssaya	98.10924369747
002B	92.64705882352
002F	92.85714285714
003D	92.85714285714
0D82	92.64705882352
0D8D	91.80672268907
0D9A.half	93.69747899159
0D9B0DCA	93.06722689075
0D9B0DD2	94.32773109243
0D9B0DD3	93.27731092436
0DA00DCA	91.59663865546
0DA00DD2	92.22689075630
0DA00DD3	91.38655462184

0DA10DD2	93.27731092436
0DA10DD3	90.54621848739
0DA20DCA	94.74789915966
0DA20DD2	92.22689075630
0DA20DD3	90.54621848739
0DA70DD2	93.48739495798
0DA70DD3	94.32773109243
0DA90DCA	93.90756302521
0DA90DD2	93.27731092436
0DA90DD3	92.64705882352
0DAD.half	93.48739495798
0DAF.rakar	91.59663865546
0DAF0DD2	93.27731092436
0DAF0DD3	92.85714285714
0DAF0DD4	93.27731092436
0DAF0DD6	93.27731092436
0DB00DCA	94.32773109243
0DB00DD3	94.95798319327
0DB1.half	93.48739495798
0DB30DD2	90.96638655462
0DB30DD3	91.17647058823
0DB30DD4	91.38655462184
0DB30DD6	91.80672268907
0DB60DD2	94.32773109243
0DB60DD3	92.85714285714
0DB80DCA	94.32773109243
0DB80DD2	94.53781512605
0DB80DD3	94.53781512605
0DB90DCA	94.32773109243
0DB90DD3	94.32773109243
0DBB0DCA	90.33613445378
0DBB0DD0	94.32773109243
0DBB0DD2	93.06722689075
0DBB0DD3	91.38655462184

0DC00DD2	94.32773109243
0DC00DD3	94.32773109243
rakar	94.53781512605
repaya	94.74789915966
0021	86.13445378151
0D83	89.91596638655
0DA10DCA	89.49579831932
0DAC0DCA	86.34453781512
0DAC0DD2	89.49579831932
0DAC0DD3	88.86554621848
0DAE0DD2	86.34453781512
0DAE0DD3	86.34453781512
0DAF0DD0	87.60504201680
003B	81.09243697478
005D	76.47058823529
00D7	75.21008403361

00F7	80.67226890756
0D9A0DCA200D0DC2	80.46218487394
0D9E	82.98319327731
0D9E0DCA	78.57142857142
0DA80DD2	78.36134453781
0DA80DD3	77.52100840336
0DAA0DD2	75.42016806722
0DAF.yansaya	84.45378151260
0DAF0DD8	76.05042016806
0DB30DD0	82.77310924369
0DBA.reph	77.52100840336
201D	80.25210084033
0D8F	10.50420168067
0D90	7.983193277310
0DA3	14.07563025210
0DA6	0.0

We observe that certain letters, such as “kandaja naasikya” ඬ (82.9%), “keti eru” ඍ (91.8%), “visargaya” ඃ (92.6%), and “anuswaraya” ං (89.9%), exhibit relatively lower presence rates within the dataset. Additionally, several glyphs are conspicuously absent from the majority of fonts analyzed. For example, “keti elu” ඏ and “diga elu” ඐ are present in only a few fonts, with a mere 10% frequency, likely reflecting their lack of usage in contemporary Sinhala (Disanayaka, 2006; Coperahewa, n.d.). Similarly, “sanyaka ja” ඥ doesn’t appear in legacy fonts. However, it’s worth noting that many legacy fonts feature a special glyph, known as the “knot of the eye”, used to form sanyaka letters by combining it with another glyph. Thus, the presence of “sanyaka ja” ඥ can be achieved through the combination of the knot of the eye and “alpapraana ja” ජ. “Mahapraana ja” ඣ also exhibits a usage percentage of 14%. Furthermore, given that the majority of legacy fonts include a glyph for the half ka (0D9A.half) with a frequency of 93.67%, forming this glyph through the combination of two glyphs is relatively straightforward.

3.1 Glyph Classification and Analysis

After identifying the most frequently used glyphs in legacy fonts, the next step is to group them for further analysis. We propose six distinct groups. These groups aim to streamline the analysis process and facilitate a deeper understanding of the glyph distribution within the dataset. The proposed groups are as follows:

- Sinhala Base Letters
- Signs (Pillam)
- Sinhala Sub Letters
- Numbers
- Punctuations
- English Letters

While our primary focus remains only on the analysis and grouping of Sinhala glyphs within legacy fonts. Yet, it's essential to acknowledge the necessity of including English letters in Unicode compliant fonts due to their role in facilitating multilingual communication.

3.1.1 Sinhala Base Letters

The base letters can be grouped into 8 groups depending on their visual similarities. Contrast to the groupings of samarawickrama, our goal is to make these groups generic as much as possible.

- Type 1: ඔ shaped glyphs (ascending glyphs)
- Type 2: ඵ shaped glyphs (base glyphs which starts from Intermediate line and ends in Intermediate line
- Type 3: ශ shaped glyphs (base glyphs which starts from Intermediate line and ends in middle line)
- Type 4: ඍ shaped glyphs (base glyphs which starts from Intermediate line and ends in bottom line)

- Type 5: ත shaped glyphs (base glyphs which starts from Intermediate line and ends in middle line)
- Type 6: ර shaped glyphs
- Type 7: Glyphs that doesn't fall under previous groups
- Type 8: Conjunct glyphs

Table 3.3: The eight types of Sinhala base letters

Type 1	0D9E, 0D91, 0D94, 0DA0, 0DA7, 0DA8, 0DA9, 0DAA, 0DAE, 0DB0, 0DB5, 0DB8, 0DB9, 0DC0, 0D9B, 0DB6, 0DAC, 0DC50DD4	ඞ, එ, ඔ, ච, ට, ඨ, ඩ, ඪ, ථ, ධ, එ, ම, ඹ, ව, ඛ, ඛ, ඞ, ඒ
Type 2	0D9D, 0DA1, 0DA2, 0DB4, 0DBA, 0DC2, 0DC3, 0D8F, 0DA6, 0D90, 0D8D	ඝ, ඡ, ජ, ප, ය, ෂ, ස, ඵ, ඡ, ඵ, සa
Type 3	0D9C, 0D9F, 0DB7, 0DC1, 0DC4	ග, හ, භ, ශ, භ
Type 4	0DAF, 0DB3	ද, ඳ
Type 5	0D9A, 0DAD, 0DB1	ක, ත, න
Type 6	0DBB, 0D8A	ර, ඊ
Type 7	0D8B, 0D85, 0D89, 0DAB, 0DBD, 0DC5, 0DC6, 0D82, 0D83	උ, අ, ඉ, ණ, ල, ළ, ෆ, ◌ං, ◌ඃ
Type 8	0DA3, 0DA4, 0DA5	ඣ, ඤ, ඥ

3.1.1.1 Type 1 Analysis

On the initial analysis we learnt that the glyphs in type 1 can further categorized into 3 sets.

Table 3.4: Sub sets of Sinhala base letters Type 1

Set 1	0D9E, 0D91, 0D94, 0DA0, 0DA7, 0DA8, 0DA9, 0DAA, 0DAE, 0DB0, 0DB5, 0DB8, 0DB9, 0DC0	Start in intermediate line
Set 2	0D9B, 0DB6	Start in middle line
Set 3	0DAC, 0DC50DD4	Can be derived from glyphs in Set 1

Type 1 – Set 1

Through many trails and errors, we identified by separating the glyph 1/3 of its total height horizontally from the bottom would successfully divide the glyph into 3 distinct parts. These parts are named as Bottom, Top Left and Top Right. However, a straight horizontal line would not clearly separate the bottom and top parts. In some situations, depending on the design of the glyph. Some parts that should belong to the top would come to bottom or some parts that should belong to the bottom would come to top part. In these cases, our algorithm would successfully distinguish the parts using connected components and allocate them in the necessary sections.

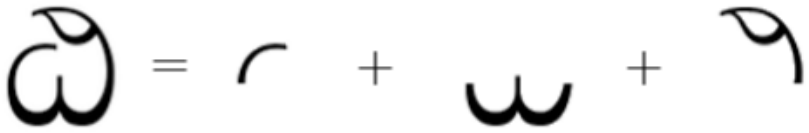


Figure 3.1: How glyph 0DAB is divided

Subsequently, each of the parts in a section is compared with all the glyphs of the same font. And the similarity will be calculated as a percentage. Also, a visual representation of the comparison would be given as shown in figure 3.2 and 3.3

Figure 3.2: 0D9E compared with 0DA0
(similarity: 66.04166667%)

Figure 3.3: 0D9E compared with 0DA9
(similarity: 97.91666667%)

Here we would categorize for each combination how many of the fonts had a similarity percentage in following ranges.

- Above 95
- 90-95
- 85-90
- 80-85
- 75-80
- Below 75

Depending on these values, the rules for which part of glyph can substitute other glyphs will be formed. The findings of this set are as following.

Bottom:

- Single curve (C1S1): 0D91, 0DA0, 0DA7, 0DAE, 0DB5, 0DB8, 0DC0
- Double curve (C1S2): 0D94, 0D9E, 0DA8, 0DA9, 0DAA, 0DB0
- Double curve with hump (C1S3): 0DB9

Top Left:

- No eye (C2S1): 0DA8, 0DAE, 0DB0, 0DA7
- Round eye (C2S2): 0D9E
- Eye (C2S3): 0DB5, 0DC0
- Eye with horizontal line (C2S4): 0D91, 0DA0, 0DA9, 0DAA
- Eye with hook 1 (C2S5): 0D94, 0DB9
- Eye with hook 2 (C2S5): 0DB8

Even though all 0D94, DB9, 0DB8 have eye with hook, 0DB8 “ma” ◌ has a longer hook, in some fonts, when compared to 0DB9 “mba” ◌ and 0D94 “o” ◌ because both these glyphs have double curve while 0DB8 has more space below the eye. We will consider 0D94, 0DB9, 0DB8 belongs to same class “Eye with hook” (C2S5) for this research.

Top Right:

- Ascending stroke (C3S1): 0D94, 0D9E, 0DA0, 0DA7, 0DA9, 0DB0, 0DB8, 0DB9, 0DC0
- Ascending end loop (C3S2): 0D91, 0DA8, 0DAA, 0DAE, 0DB5

The glyphs in type 1 set 1 can be expressed using relevant parts as:

- 0D91 = C1S1 + C2S4 + C3S2
- 0DA0 = C1S1 + C2S4 + C3S1
- 0DA7 = C1S1 + C2S1 + C3S1
- 0DAE = C1S1 + C2S1 + C3S2
- 0DB5 = C1S1 + C2S3 + C3S2
- 0DB8 = C1S1 + C2S5 + C3S1
- 0DC0 = C1S1 + C2S3 + C3S1
- 0D94 = C1S2 + C2S5 + C3S1
- 0D9E = C1S2 + C2S2 + C3S1
- 0DA8 = C1S2 + C2S1 + C3S2
- 0DA9 = C1S2 + C2S4 + C3S1
- 0DAA = C1S2 + C2S4 + C3S2
- 0DB0 = C1S2 + C2S1 + C3S1
- 0DB9 = C1S3 + C2S5 + C3S1

Among these glyphs 0DA0, 0DA7, 0DAE, 0DB8, 0DC0, 0D94, 0D9E, 0DA8, 0DA9, 0DB0, 0DB9 can be considered as the candidates for determining the minimum number of glyphs needed to be created for this set. since these are the glyphs that has a lesser chance of having interactions with top left and right sides. the glyphs ending with a loop has a tendency to touch the right-side eye.

6 glyphs were identified as the minimum glyph count which needs to create all 14 glyphs in type 1 set 1 by the algorithm. And 4 different glyphs sets are suggested

Table 3.5: Different ways of glyphs to be designed for Type 1-Set 1

Code	*Class*	*Glyph*
0DC0	C1S1, C2S3, C3S1	ව
0D94	C1S2, C2S5	ඹ
0D9E	C2S2	ඩ
0DA8	C2S1, C3S2	ය
0DA9	C2S4	ඩ
0DB9	C1S3	ඹ

Code	*Class*	*Glyph*
0D9E	C1S2, C2S2, C3S1	ඩ
0DA8	C2S1, C3S2	ය
0DA9	C2S4	ඩ
0DB9	C1S3, C2S5	ඹ
0DA0	C1S1	ච
0DC0	C2S3	ව

Code	*Class*	*Glyph*
0DA8	C1S2, C2S1, C3S2	ය
0DA9	C2S4, C3S1	ඩ
0DB9	C1S3, C2S5	ඹ
0DA0	C1S1	ච
0DC0	C2S3	ව
0D9E	C2S2	ඩ

Code	*Class*	*Glyph*
0DB0	C1S2, C2S1, C3S1	ධ
0DB9	C1S3, C2S5	ඹ
0DA0	C1S1, C2S4	ච
0DAE	C3S2	ථ
0DC0	C2S3	ව
0D9E	C2S2	ඩ

Type 1-Set 2

This set only contains 2 glyphs 0DB6 “ba” බ and 0D9B “kha” ඛ. at the first glance it seems that we can split the glyph vertically into 2 parts. But with our initial analysis of the glyph, we came to the conclusion that finding the exact place to split the glyphs vertically is a challenge so a decision was made to split the glyph horizontally into 2 parts like we did in Set 1. This resulted in 2 sections Bottom and Top. As same as for type 1 the split position was 1/3 of the total height of the glyph from bottom.

Figure 3.4: How glyph 0DB6 is divided

Since there are only 2 glyphs only a single set was generated and the it is obvious that the bottom part is not substitutable while the top part is substitutable.

Figure 3.5: 0D9B compared with 0DB6
(Similarity: 94.66230937%)

Figure 3.6: 0D9B compared with 0DB6
(Similarity: 98.08306709%)

Type 1-Set 3

This set also contains only 2 glyphs 0DAC “nndda” ඬ and 0DC50DD4 “lu” ළු unlike in previous situations the glyphs can be substitute by 2 glyphs in the set 1 completely while only remaining a minor change to be done. 0DC50DD4 ළු is however little different since some fonts have a different style for this letter

Figure 3.7: Different styles of glyph “ළු”
Fonts: FM-Bindumathi and DL-Ridhma

0DAC can be substitute by 0DA9 and 0DC50DD4 can be substitute by 0DB5. we suggest the usage of existing glyph and adding a knot to generate the glyphs in set 3. However, the use of Bottom and Top Left portions of relevant glyphs and let the user decide how the eye is to be designed is also considered valid.

Figure 3.8: 0DAC with bottom and top right parts removed using 0DA9

Figure 3.9: 0DC50DD4 with bottom and top right parts removed using 0DB5

3.1.1.2 Type 2 analysis

On the initial analysis we learnt that the glyphs in type 2 can further categorized into 3 sets.

Table 3.6: Sub sets of Sinhala base letters Type 2

Set 1	0D9D, 0DA1, 0DA2, 0DB4, 0DBA, 0DC2, 0DC3	
Set 2	0D8F, 0DA6	Can be derived from glyphs in Set 1
Set 3	0D90, 0D8D	Can be derived from glyphs in Set 1, 2 and pillam

Type 2 – Set 1

These characters fall in between lines 2 and 4 in 5 reference lines hence the ideal position for horizontally splitting the glyph would be ¼ of the height of the glyph. But doing so sometimes would not correctly split the glyph. Hence splitting at ½ of the glyph height is chosen for this set. The glyphs will be split into 3 sections. Bottom, Top Left and Top Right. The splitting algorithm will identify and allocate if either of parts belong to top or bottom is split incorrectly. The eyes of top left and right parts being stick to together can be seen commonly in this glyph set. The algorithm is designed find the lowest position in the middle of the eyes and split vertically from there.

Figure 3.10: How glyph 0D9D is divided

This set will undergo the same comparison process and the findings for this set are as following.

Bottom:

- Single curve (C1S1): 0DA1, 0DA2, 0DB4, 0DC2
- Double curve (C1S2): 0DBA, 0DC3
- Double curve with hump (C1S3): 0D9D

Top Left:

- Eye (C2S1): 0DA1, 0DA2, 0DB4, 0DC2
- Eye with horizontal line (C2S2): 0D9D, 0DC3
- No eye (C2S3): 0DBA

Top Right:

- Eye (C3S1): 0D9D, 0DB4, 0DBA, 0DC3
- Eye with big knot (C3S2): 0DA1
- Eye with small knot (C3S3): 0DA2
- Eye with base loop (C3S4): 0DC2

The glyphs in type 2 set 1 can be expressed using relevant parts as:

- 0D9D = C1S3 + C2S2 + C3S1
- 0DA1 = C1S1 + C2S1 + C3S2
- 0DA2 = C1S1 + C2S1 + C3S3
- 0DB4 = C1S1 + C2S1 + C3S1
- 0DBA = C1S2 + C2S3 + C3S1
- 0DC2 = C1S1 + C2S1 + C3S4
- 0DC3 = C1S2 + C2S2 + C3S1

All glyphs 0D9D, 0DA1, 0DA2, 0DB4, 0DBA, 0DC2, 0DC3 are candidates for determining the minimum number of glyphs needed to be created for this set. The algorithm identified 5 glyphs is the minimum glyph count that will be needed to create the 7 glyphs in Type 2-set 1. And suggested a way to do this

Table 3.7: Different ways of glyphs to be designed for Type 2-Set 1

Code	**Class**	**Glyph**
0D9D	C1S3, C2S2, C3S1	ඝ
0DA1	C1S1, C2S1, C3S2	ඡ
0DA2	C3S3	ජ
0DBA	C1S2, C2S3	ය
0DC2	C3S4	ෂ

Type 2-Set 2

This set contains 2 glyphs 0D8F “I” ඏ and 0DA6 “nyja” ඦ. These 2 glyphs are almost none existent in legacy fonts. this might be due to the fact that they can be derived from 0DB4 “pa” ප and 0DA2 “ja” ජ by changing the eye of the glyph to a knotted eye. 0DA6 was not found on any of the legacy fonts we were testing 0D8F was found on 50 fonts. this is how the font looks like when 0DB4 is removed from 0D8F

Figure 3.11: 0D85 with parts removed using 0DB4

Type 2-Set 3

This set contains 2 glyphs 0D90 “II” ඐ and 0D8D “r” ඍ. 0D90 is created by combining 0D8F “I” ඏ and 0DDF (gayanukitta) ෟ and 0D8D is created by combining 0DC3 “sa” ස and 0DD8 (gatapilla) ෘa. We can see that in some fonts the eye is constructed with a horizontal line while in some fonts the horizontal line cannot be found.

3.1.1.3 Type 3 analysis

This type contains the 5 glyphs 0D9C, 0D9F, 0DB7, 0DC1, 0DC4. All glyphs in type 1 and type 2 were split horizontally. But for type 3, the most suitable way separating the glyph would be to split the glyph vertically at 2 points. Thus, splitting the glyph into 3 parts. We name them as Left, Middle and Right.

Figure 3.12: How glyph 0D9C is divided

Findings for this type are as following.

Left:

- Head (C1S1): 0D9C,0DC1
- Head with shoulder joint (C1S2): 0D9F
- Head with eye (C1S3): 0DC4
- Head with horizontal lined eye (C1S4): 0DB7

Here due to the extreme similarities of 0DC4 and 0DB7 it gives a high similarity percentage when checking the similarities as shown in figure 3.13. Figure 3.14 and 3.15 shows 2 examples of this. but we would consider them as different sets since these are not actually substitutable by each other.

Above 95	34
90-95	166
85-90	85
80-85	62
75-80	44
Below 75	21

Figure 3.13: Similarities of 0DC4 and 0DB7 in different fonts as percentages

Figure 3.14: FM-Bindumathi
(Similarity: 94.66230937%)

Figure 3.15: DL-Ridhma
(Similarity: 93.34719335%)

Middle:

- Curve to curve joint (C2S1): 0D9C, 0D9F, 0DB7, 0DC4, 0DC1

Right:

- Tail (C3S1): 0DB7, 0DC4, 0D9F, 0D9C
- Tail with base loop(C3S2): 0DC1

The glyphs in type 1 set 2 can be expressed using relevant parts as:

- 0D9C = C1S1 + C2S1 + C3S1
- 0D9F = C1S2 + C2S1 + C3S1
- 0DB7 = C1S4 + C2S1 + C3S1
- 0DC1 = C1S1 + C2S1 + C3S2
- 0DC4 = C1S3 + C2S1 + C3S1

All glyphs in this type are candidates for determining the minimum number of glyphs needed to be created for this type. The algorithm identified 4 glyphs is the minimum glyph count that will be needed to create the 5 glyphs in Type 3. And suggested 3 ways to do this.

Table 3.8: Different ways of glyphs to be designed for Type 3

Code	*Class*	*Glyph*
0D9F	C1S2, C2S1, C3S1	ඟ
0DB7	C1S4	ඹ
0DC1	C1S1, C3S2	ශ
0DC4	C1S3	හ

Code	*Class*	*Glyph*
0DB7	C1S4, C2S1, C3S1	ඹ
0DC1	C1S1, C3S2	ශ
0DC4	C1S3	හ
0D9F	C1S2	ඟ

Code	*Class*	*Glyph*
0DC1	C1S1, C2S1, C3S2	ශ
0DC4	C1S3, C3S1	හ
0D9F	C1S2	ඟ
0DB7	C1S4	ඹ

3.1.1.4 Type 4 analysis

This type only contains 2 glyphs 0DAF “da” ද and 0DB3 “nda” ඳ and the analysis of glyphs shows that 0DB3 is constructed by adding a shoulder joint / eye knot to the glyph 0DAF. But some glyphs also change the eye hence designing these as 2 glyphs is ideal.

Figure 3.16: 0DB3 with parts removed using 0DAF

3.1.1.5 Type 5 analysis

This type consists of 3 glyphs 0D9A “ka” ක, 0DAD “ta” ත, 0DB1 “na” න. In all these glyphs we can find a cross road like feature as shown in figure 3.17. This can be identified as a split point where the glyphs can be divided into 3 parts. We can identify them as Top, Bottom and right.

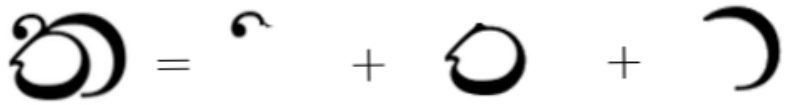


Figure 3.17: How glyph 0DAD is divided

Once we analyze the divided glyphs font wise, we can notice some things that were not available in other types we analyzed before.

- The similarity of bottom part is lover than 75% in most of the glyphs meaning they are not substitutable.
- The similarity of the eye seems to be around 85-90% in most fonts. but the eyes are being so small leads to the percentages varying drastically even for a small difference of a pixel.
- The tail seems to be similar 90% or above all the glyphs

Since the top part and right part are both similar and can be substituted for each glyph, we can select 0DB1 as the glyph to be designed. The main reason for choosing 0DB1 is because the part under the eye is curvy in this glyph. hence it has a low chance of the eye being connected to the bottom part when compared with other 2 glyphs in this group. This is shown in figure 3.18.

Figure 3.18: How eye collide with the body

3.1.1.6 Type 6 analysis

This type also has only 2 glyphs 0DBB “ra” ර and 0D8A “ii” ඊ. The analysis shows that 0D8A can be derived by adding 2 circles to the sides of diagonal stroke of glyph 0DBB.

Figure 3.19: 0D8A with parts removed using 0DBB

3.1.1.7 Type 7 analysis

For this type we were unable to find any close resemblances with other glyphs. It consists of 7 glyphs.

- 0D8B උ
- 0D85 අ
- 0D89 ඉ
- 0DAB ණ
- 0DBD ල
- 0DC5 ළ
- 0DC6 ෆ
- 0D82 ං
- 0D83 ඃ

Even if it seems that 0D8B, 0DBD, 0DC5 have similar properties to the naked eye, our initial analysis shows that they are infect not similar. Hence the design of these glyphs is required. The same is applicable to 0D82 and 0D83.

3.1.1.8 Type 8 analysis

The glyphs in this type are all conjunct letters. According to the book Sinhala Graphology (Disanayaka, 2006) these are letters like conjunct letters. This type consists of 3 glyphs.

- 0DA3 ඣ
- 0DA4 ඤ
- 0DA5 ඥ

0DA3 ඣ is created by combining the 2 glyphs 0D9A “ka” ක and 0DB0 “dha” ධ. The tail of 0D9A is connected to the start of 0DBO and the bodies of two glyphs are stick together. This can be generated using the glyphs we have. But letting the font developer design the exact way the tail and front connects would be ideal.

Figure 3.20: How glyph 0DA3 is created using 0D9A and 0DB0

0DA4 ඤ is created by combining the 2 glyphs 0D9A “ka” ක and 0DAF “da” ද. The tail of 0D9A is connected to the eye of 0DAF and the bodies of two glyphs are stick together just like with the glyph 0DA3. We can also see a knotted eye for 0D9A in this glyph hence the knot in the eye and the tail should be designed by the font developer.

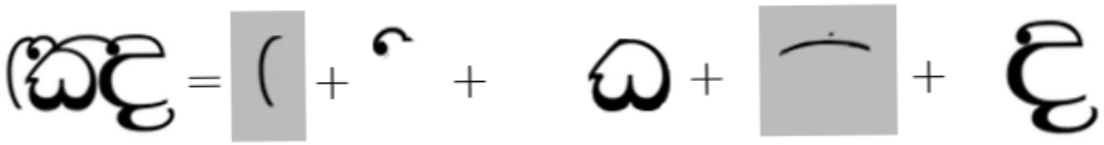


Figure 3.21: How glyph 0DA4is created using 0D9A and 0DAF

0DA5 ඥ is created by combining the 2 glyphs 0D9F “nnga” ඟ and 0DAF “da” ද. The tail of 0D9F is connected to the eye of 0DAF and the bodies of two glyphs are stick together just like with the glyph 0DA3. As with all the glyphs in this type the tail connecting two glyphs should be designed by the font developer.

Figure 3.22: How glyph 0DA5is created using 0D9F and 0DAF

3.1.1.9 Conclusion for Sinhala Base letters

For the 61 main letters in Sinhala Unicode, we can see that 11 of these letters are formed by adding a sign to an existing letter, we will discuss about these in coming chapters. 4 letters that are formed by doing modifications to existing letter and 3 are formed by connecting 2 glyphs. And also, a letter which is not considered as a separate letter but has a separate glyph.

So, in order to generate glyphs for the 61 letters we need to create minimum of 31 glyphs and 9 glyphs with minor modifications should be created.

3.1.2 Signs (Pillam)

We can find 16 signs in legacy fonts. they are,

Table 3.9: Signs/Pillam found in Sinhala legacy and Unicode fonts

Name/Codepoint	*Description*	*glyph*	*Availability in legacy fonts*
0DCA	al-lakuna	්	Yes
0DCF	aela-pilla	ා	Yes
0DD0	keti aeda-pilla	ැ	Yes
0DD1	diga aeda-pilla	ෑ	Yes
0DD2	keti is-pilla	ි	Yes
0DD3	diga is-pilla	ී	Yes
0DD4	keti paa-pilla	ු	Yes
.0DD4	keti vak paa-pilla		Yes
0DD6	diga paa-pilla	ූ	Yes
.0DD6	diga vak paa-pilla		Yes
0DD8	gaetta-pilla	ෘ	Yes
0DD9	kombuwa	ෙ	Yes
0DDF	gayanukitta	ෟ	Yes
rakar	rakaranshaya		Yes

reph	repaya	◌්	Yes
yanssaya	yanssaya		Yes
0DF3	diga gayanukitta	◌ෟ	No
0DF4	kundaliya	෴	No

0DF2 “diga gayanukitta” is not available in any of the legacy fonts but is available with Unicode fonts so we would consider them as a glyph to be designed. 0DF4 “kundaliya” is the only punctuation mark listed for Unicode in Sinhala but rather than categorizing it under puncturing marks we will categorize it under this set.

We can notice that the glyphs 0DCA, 0DD2, 0DD3, 0DD4, 0DD6 and repaya appear multiple times in same legacy font. This was due to the fact that Sinhala letters are of different widths and heights so to adopt accordingly multiple glyphs are designed and handled using substitution rules. The font developer would have to design 18 separate glyphs for all these signs.

Once they are designed the following glyphs would be able to be generated using them.

Table 3.10: Signs/Pillam that can be generated using other glyphs

Name/Codepoint	*Description*	*glyph*
0DF2	diga gaetta-pilla	◌ාා
0DDA	diga kombuwa	◌ෙ
0DDB	kombu deka	◌ෙෙ
0DDC	kombuwa haa aela-pilla	◌ො
0DDD	kombuwa haa diga aela-pilla	◌ෞ
0DDE	kombuwa haa gayanukitta	◌ෞ

3.1.3 Sinhala Sub letters

These glyphs are formed by adding a sign to an existing glyph. When adding a sign to a letter, we can see that there are 2 behaviors. One set will combine the glyph with the glyph of the sign and it will not modify the shape of the existing but rather extend the glyph. The other set will change the shape of the glyph. This feature was well utilized in legacy fonts. since they only had a limited number of codepoints available for the glyphs to be mapped, the font designers only designed the glyphs that the appearance is changed when the sign is applied. For the other glyphs they simply designed the sign only.

Figure 3.23: How glyph change with different signs
Original glyph | The upper part of the glyph is changed | Glyph is extended but original shape remains the same

When doing this the designer had to be precise when designing all the glyphs so that for any glyph that will combine the sign will match perfectly forming a visually seamless letter. This was a tedious process so later the developers tend to use anchor points so that rather than designing all the glyphs to comply a sign it was easy to define a position where the sign should be aligned with that glyph.

Defining anchor points also take time so if the designer could design a glyph with all its variants with different signs it would be much easier for the developer. But then again, the number of glyphs need to be design will greatly increase. This is an aspect we are going to give a solution through research.

here in table 3.11, we will show how glyphs in sub category are designed for each type of glyphs in the main category.

Table 3.11: Sub glyphs with their main glyph

Type	*Glyph*	*Sub Glyph/s*
Type 1	0D9E	0D9E0DCA, 0D9E0DD2, 0D9E0DD3
	0D91	0D92
	0D94	0D95

	0DA0	0DA00DCA, 0DA00DD2, 0DA00DD3
	0DA7	0DA70DCA, 0DA70DD2, 0DA70DD3
	0DA8	0DA80DD2, 0DA80DD3
	0DA9	0DA90DCA, 0DA90DD2, 0DA90DD3
	0DAA	0DAA0DD2, 0DAA0DD3
	0DAE	0DAE0DD2, 0DAE0DD3
	0DB0	0DB00DCA, 0DB00DD2, 0DB00DD3
	0DB5	0DB50DD2, 0DB50DD3
	0DB8	0DB80DCA, 0DB80DD2, 0DB80DD3
	0DB9	0DB90DCA, 0DB90DD2, 0DB90DD3
	0DC0	0DC00DCA, 0DC00DD2, 0DC00DD3
	0D9B	0D9B0DCA, 0D9B0DD2, 0D9B0DD3
	0DB6	0DB60DCA, 0DB60DD2, 0DB60DD3
	0DAC	0DAC0DCA, 0DAC0DD2, 0DAC0DD3
	0DC50DD4	-
Type 2	0D9D	-
	0DA1	0DA10DCA, 0DA10DD2, 0DA10DD3
	0DA2	0DA20DCA, 0DA20DD2, 0DA20DD3
	0DB4	-
	0DBA	0DBA.reph
	0DC2	-
	0DC3	-
	0D8F	-
	0DA6	-
	0D90	-
	0D8D	-
Type 3	0D9C	-
	0D9F	-
	0DB7	-
	0DC1	0DC10DD3.rakar
	0DC4	-
Type 4	0DAF	0DAF.rakar, 0DAF.yansaya, 0DAF.yansaya0DCF, 0DAF0DCF, 0DAF0DD0, 0DAF0DD1, 0DAF0DD2, 0DAF0DD3, 0DAF0DD4, 0DAF0DD6, 0DAF0DD8, 0DAF0DDD

	0DB3	0DB3.yansaya, 0DB3.yansaya0DCF, 0DB30DCF, 0DB30DD0, 0DB30DD1, 0DB30DD2, 0DB30DD3, 0DB30DD4, 0DB30DD6, 0DB30DDD
Type 5	0D9A	-
	0DAD	-
	0DB1	-
Type 6	0D8A	-
	0DBB	0DBB0DCA, 0DBB0DD0, 0DBB0DD1, 0DBB0DD2, 0DBB0DD3
Type 7	0D8B	-
	0D82	-
	0D83	-
	0D85	-
	0D89	-
	0DAB	0DAB.reph, 0DAB0DD2, 0DAB0DD3
	0DBD	0DBD0DD4, 0DBD0DD6
	0DC5	-
	0DC6	-
Type 8	0DA3	-
	0DA4	-
	0DA5	-

3.1.3.1 Type 1 analysis

The main part which is changed in this seems to be the Top right section of the glyph Top left and Bottom parts remain intact. Analysis of the top part showed that it can be categorized into 5 groups.

- Rehana
- Keti is-pilla with ascending stroke
- Keti is-pilla with ascending end loop
- Diga is-pilla with ascending stroke
- Diga is-pilla with ascending end loop

Including these 5 types our findings in Type 1 can be amended as following.

Bottom:

- Single curve (C1S1): 0D91, 0DA0, 0DA7, 0DAE, 0DB5, 0DB8, 0DC0, 0DA00DCA, 0DA00DD2, 0DA00DD3, 0DA70DCA, 0DA70DD2, 0DA70DD3, 0DAE0DD2, 0DAE0DD3, 0DB50DD2, 0DB50DD3, 0DB80DCA, 0DB80DD2, 0DB80DD3, 0DC00DCA, 0DC00DD2, 0DC00DD3

- Double curve (C1S2): 0D94, 0D9E, 0DA8, 0DA9, 0DAA, 0DB0, 0D95, 0D9E0DCA, 0D9E0DD2, 0D9E0DD3, 0DA80DD2, 0DA80DD3, 0DA90DCA, 0DA90DD2, 0DA90DD3, 0DAA0DD2, 0DAA0DD3, 0DB00DCA, 0DB00DD2, 0DB00DD3

- Double curve with hump (C1S3): 0DB9, 0DB90DCA, 0DB90DD2, 0DB90DD3

Top Left:

- No eye (C2S1): 0DA8, 0DAE, 0DB0, 0DA7, 0DA70DCA, 0DA70DD2, 0DA70DD3, 0DA80DD2, 0DA80DD3, 0DAE0DD2, 0DAE0DD3, 0DB00DCA, 0DB00DD2, 0DB00DD3

- Round eye (C2S2): 0D9E, 0D9E0DCA, 0D9E0DD2, 0D9E0DD3

- Eye (C2S3): 0DB5, 0DC0, 0DB50DD2, 0DB50DD3, 0DC00DCA, 0DC00DD2, 0DC00DD3

- Eye with horizontal line (C2S4): 0D91, 0DA0, 0DA9, 0DAA, 0DA00DCA, 0DA00DD2, 0DA00DD3, 0DA90DCA, 0DA90DD2, 0DA90DD3, 0DAA0DD2, 0DAA0DD3

- Eye with hook (C2S5): 0D94, 0DB9, 0DB8, 0D95, 0DB80DCA, 0DB80DD2, 0DB80DD3, 0DB90DCA, 0DB90DD2, 0DB90DD3

Top Right:

- Ascending stroke (C3S1): 0D94, 0D9E, 0DA0, 0DA7, 0DA9, 0DB0, 0DB8, 0DB9, 0DC0

- Ascending end loop (C3S2): 0D91, 0DA8, 0DAA, 0DAE, 0DB5

- Rehana (C3S3): 0D95, 0D9E0DCA, 0DA00DCA, 0DA70DCA, 0DA90DCA, 0DB00DCA, 0DB80DCA, 0DB90DCA, 0DC00DCA
- Keti is-pilla with ascending stroke (C3S4): 0D9E0DD2, 0DA00DD2, 0DA70DD2, 0DA90DD2, 0DB00DD2, 0DB80DD2, 0DB90DD2, 0DC00DD2
- Keti is-pilla with ascending end loop (C3S5): 0DA80DD2, 0DAA0DD2, 0DAE0DD2, 0DB50DD2
- Diga is-pilla with ascending stroke (C3S6): 0D9E0DD3, 0DA00DD3, 0DA70DD3, 0DA90DD3, 0DB00DD3, 0DB80DD3, 0DB90DD3, 0DC00DD3
- Diga is-pilla with ascending end loop (C3S7): 0DA80DD3, 0DAA0DD3, 0DAE0DD3, 0DB50DD3

This gives a total of 47 glyphs. The amended glyphs in type 1 can be expressed using relevant parts as:

- 0D91 = C1S1 + C2S4 + C3S2
- 0DA0 = C1S1 + C2S4 + C3S1
- 0DA7 = C1S1 + C2S1 + C3S1
- 0DAE = C1S1 + C2S1 + C3S2
- 0DB5 = C1S1 + C2S3 + C3S2
- 0DB8 = C1S1 + C2S5 + C3S1
- 0DC0 = C1S1 + C2S3 + C3S1
- 0DA00DCA = C1S1 + C2S4 + C3S3
- 0DA00DD2 = C1S1 + C2S4 + C3S4
- 0DA00DD3 = C1S1 + C2S4 + C3S6
- 0DA70DCA = C1S1 + C2S1 + C3S3
- 0DA70DD2 = C1S1 + C2S1 + C3S4

- 0DA70DD3 = C1S1 + C2S1 + C3S6
- 0DAE0DD2 = C1S1 + C2S1 + C3S5
- 0DAE0DD3 = C1S1 + C2S1 + C3S7
- 0DB50DD2 = C1S1 + C2S3 + C3S5
- 0DB50DD3 = C1S1 + C2S3 + C3S7
- 0DB80DCA = C1S1 + C2S5 + C3S3
- 0DB80DD2 = C1S1 + C2S5 + C3S4
- 0DB80DD3 = C1S1 + C2S5 + C3S6
- 0DC00DCA = C1S1 + C2S3 + C3S3
- 0DC00DD2 = C1S1 + C2S3 + C3S4
- 0DC00DD3 = C1S1 + C2S3 + C3S6
- 0D94 = C1S2 + C2S5 + C3S1
- 0D9E = C1S2 + C2S2 + C3S1
- 0DA8 = C1S2 + C2S1 + C3S2
- 0DA9 = C1S2 + C2S4 + C3S1
- 0DAA = C1S2 + C2S4 + C3S2
- 0DB0 = C1S2 + C2S1 + C3S1
- 0D95 = C1S2 + C2S5 + C3S3
- 0D9E0DCA = C1S2 + C2S2 + C3S3
- 0D9E0DD2 = C1S2 + C2S2 + C3S4
- 0D9E0DD3 = C1S2 + C2S2 + C3S6
- 0DA80DD2 = C1S2 + C2S1 + C3S5

- 0DA80DD3 = C1S2 + C2S1 + C3S7
- 0DA90DCA = C1S2 + C2S4 + C3S3
- 0DA90DD2 = C1S2 + C2S4 + C3S4
- 0DA90DD3 = C1S2 + C2S4 + C3S6
- 0DAA0DD2 = C1S2 + C2S4 + C3S5
- 0DAA0DD3 = C1S2 + C2S4 + C3S7
- 0DB00DCA = C1S2 + C2S1 + C3S3
- 0DB00DD2 = C1S2 + C2S1 + C3S4
- 0DB00DD3 = C1S2 + C2S1 + C3S6
- 0DB9 = C1S3 + C2S5 + C3S1
- 0DB90DCA = C1S3 + C2S5 + C3S3
- 0DB90DD2 = C1S3 + C2S5 + C3S4
- 0DB90DD3 = C1S3 + C2S5 + C3S6

in Main Type 1-Set 1 we have a candidate glyph set that is most suitable for analyzing.

0DA0, 0DA00DCA, 0DA00DD2, 0DA00DD3, 0DA7, 0DA70DCA, 0DA70DD2, 0DA70DD3, 0DAE, 0DAE0DD2, 0DAE0DD3, 0DB8, 0DB80DCA, 0DB80DD2, 0DB80DD3, 0DC00DCA, 0DC00DD2, 0DC00DD3, 0D94, 0D95, 0D9E, 0DA8, 0DA80DD2, 0DA80DD3, 0DA90DCA, 0DA90DD2, 0DA90DD3, 0DB0, 0DB00DCA, 0DB00DD2, 0DB00DD3, 0DB9, 0DB90DCA, 0DB90DD2, 0DB90DD3

Table 3.12: Different ways of glyphs to be designed for Type 1

Code	*Class*	*Glyph*
0DC00DD3	C1S1, C2S3, C3S6	වී
0DB90DCA	C1S3, C2S5, C3S3	ඹි
0DB90DD2	C3S4	ඹී
0DA0	C2S4, C3S1	ච
0DA7	C2S1	ට
0DAE	C3S2	ඵ
0D94	C1S2	ඔ
0D9E	C2S2	ඞ
0DA80DD2	C3S5	යී
0DA80DD3	C3S7	යී

Code	*Class*	*Glyph*
0DC00DD2	C1S1, C2S3, C3S4	වි
0DC00DD3	C3S6	වී
0DB90DCA	C1S3, C2S5, C3S3	ඹි
0DA0	C2S4, C3S1	ච
0DA7	C2S1	ට
0DAE	C3S2	ඵ
0D94	C1S2	ඔ
0D9E	C2S2	ඞ
0DA80DD2	C3S5	යී
0DA80DD3	C3S7	යී

Code	*Class*	*Glyph*
0DB9	C1S3, C2S5, C3S1	ඹ
0DA80DD2	C1S2, C2S1, C3S5	යී
0DA80DD3	C3S7	යී
0DA90DCA	C2S4, C3S3	ඩි
0DA90DD2	C3S4	ඩී
0DA90DD3	C3S6	ඩී
0DA00DCA	C1S1	වි
0DC00DCA	C2S3	වි
0DAE	C3S2	ඵ

0D9E	C2S2	ඞ

The 47 glyphs can be generated by designing minimum of 10 glyphs. If we included 0D91, 0DAA, 0DB5 and their variants the minimum number of glyphs need to be designed could have been minimized to 9 but, since there's high chance of the eye touching the ascending end loop in those glyphs omitting these glyphs would ensure more accurate results.

Designing the glyphs in following order will ensure quicker designing process. Since we can utilize parts of already generated glyphs.

1. 0DA7
2. 0DAE
3. 0DA0
4. 0D94
5. 0DB90DCA
6. 0D9E
7. 0DC00DD2
8. 0DC00DD3
9. 0DA80DD2
10. 0DA80DD3

0D92 ඒ was not included in the above list due to the fact it can be created by joining the 2 glyphs 0D91 එ and 0DCA “kodiya” ්. only 25 out of 475 fonts contained a separate glyph for it.

3.1.3.2 Type 2 analysis

For type 2 there are only 7 glyphs that belongs to 3 main glyphs. 2 of the main glyphs are 0DA1 “cha” ඡ and 0DA2 “ja” ජ. These letters in Sinhala have some special features. Unlike other letters the “kodiya” is positioned at the middle of the glyph between the eyes where as in other glyphs it is usually positioned at the right side of the letter. This might be due to the fact that there’s a diagonal stroke already at the right side of the letter. Also, the “is-pilla” is designed to be aligned with the diagonal stroke of the letter rather than horizontally on top of the letter unlike in other letters. But in different fonts we can see that rule was not strictly followed. This is shown in figure 3.24.

Figure 3.24: How is-pilla and kodiya denoted in different fonts.

Hence, we will consider the glyphs 0DA10DCA and 0DA20DCA are generatable glyphs by combining 0DA1/2 and 0DCA “kodiya” ් by positioning it in the middle of the eyes. While letting the font designer design 0DA10DD2, 0DA10DD3, 0DA20DD2, 0DA20DD3.

The other glyph available in this type is 0DBA.reph ඣ which is a combination of 0DBA “ya” ය and “repaya” ්. This glyph is also not needed to be designed since it can be generated using existing glyphs.

Figure 3.25: 0DBA.reph with parts removed using 0DBA

3.1.3.3 Type 3 analysis

There’s only a single glyph under this type 0DC10DD3.rakar ශ්‍රී which is available in 55.46% of examined fonts. this glyph is a combination of 3 glyphs 0DC1 “sha” ශ 0DD3 “diga is-pilla” ී and “rakaranshaya”. glyph analysis shows that some minor changes are made for the bottom part of the glyph. But utilizing existing glyphs to generate the desired glyph also gives satisfactory results.

3.1.3.4 Type 4 analysis

This type can be considered as the most versatile sub set of glyphs. Despite only being consist of 2 main glyphs the letters ඍ and ඏ has a tradition of having the signs which are supposed to displayed to the right side of the letter are positioned below the glyph instead in the bottom line (Figure 3.26). But disanayaka explains that this practice is fading gradually while the practice of having the signs to the right side of letter as in other letters becoming more popular in the recent past (Figure 3.17).

Figure 3.26: Signs are displayed below the glyph

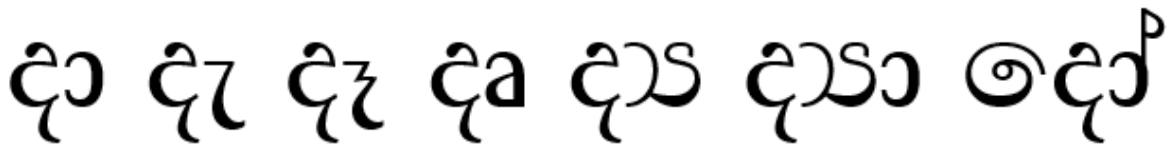


Figure 3.27: Signs are displayed alongside the glyph

Since there's a more common and simple way of presenting these glyphs, we will not design these glyphs separately. Once we omit the above glyphs, we are left with the following glyphs which are available in the legacy fonts.

- 0DAF.rakar ෟ
- 0DAF0DD2 ෛ
- 0DAF0DD3 ෞ
- 0DAF0DD4 ෟ
- 0DAF0DD6 ෟ
- 0DB30DD2 ෛ
- 0DB30DD3 ෞ

- 0DB30DD4 ෴
- 0DB30DD6 ෵

Even though we find the glyph ෴ in 91% of the examined fonts the glyph ෵ is only can be found in only 1% of the fonts. However, we would consider this as a glyph to be designed by modifying 0DAF.rakar

The width of the two letters in type 4 is narrower compared to the other glyphs. Hence, we have to design the glyphs for “is-pilla” and “diga is-pilla” to be narrower than the general glyphs for them as shown in Figure 3.28. The design of 0DAF0DD2 and DAF0DD3 with narrower “is-pilla” and “diga is-pilla” is possible. Hence creating separate glyphs is not necessary because of this. So, we would have to design 3 glyphs for this sub type and the other 3 glyphs can be generated using the designed glyphs. A total of 8 glyphs should be designed for this type to generate the total 12 glyphs for this type.

Figure 3.28: Length difference of is-pilla

3.1.3.5 Type 5 analysis

There are no glyphs for this set.

3.1.3.6 Type 6 analysis

Even though type 6 base letters had 2 glyphs one of them was a vowel and no signs are applied to it. But for the glyph 0DBB we can find 5 sub glyphs. The rules for this letter are also bit different when compared to others. Here also the “kodiya” ◌් is positioned at the left side of the glyph and also the “is-pilla” and “diga is-pilla” is aligned diagonally with the stroke at the top. Sometimes we can notice some fonts have done slight changes to the diagonal stroke too. Also, to represent the vowel sound “ae” and “aee” sounds the glyphs ඇ and ඈ used respectively. The signs “aedaya” ◌ෙ and “diga aedaya” ◌ේ which is used to represent the vowel sound “ae” and “aee” are used to represent vowel sounds “u” and “uu” in this letter. So, the glyphs that needs to be designed for this type are:

- 0DBB0DCA ර්
- 0DBB0DD0 රැ
- 0DBB0DD1 රෑ
- 0DBB0DD2 රි
- 0DBB0DD3 රී

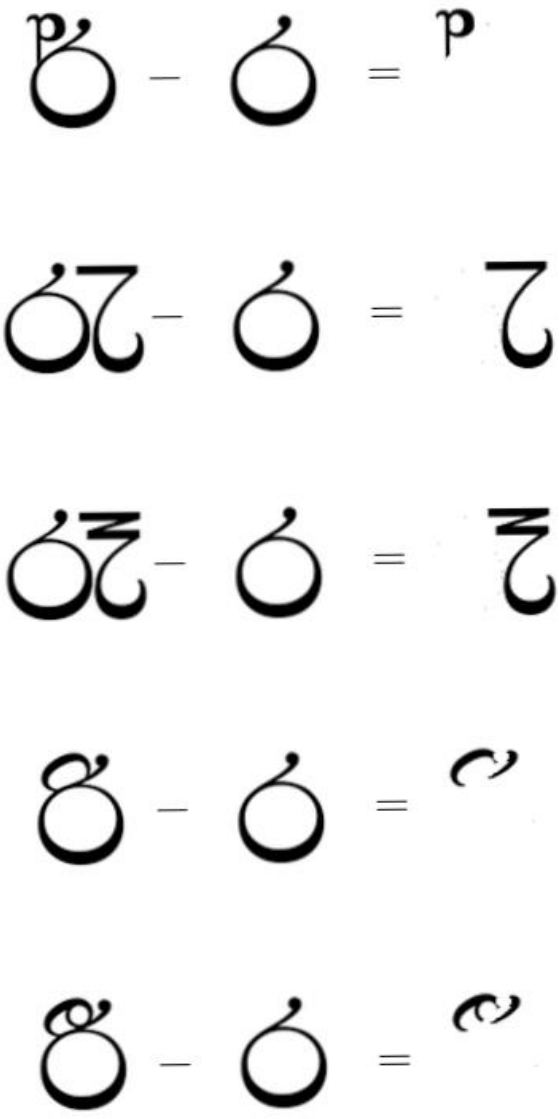


Figure 3.29: How sub glyphs for 0DBB are formed with signs

Here, Since the base glyph is already designed, only minor design changes are left to be done for the font developer. And since for glyph 0DBB0DCA only the position of the “kodiya” is to be determined it can be generated automatically using glyphs 0DBB ර and 0DCA ්.

For the type 6, there are 7 glyphs including both base and sub letters. In order to design these glyphs 1 whole glyph should be designed and other 5 glyphs can be designed by doing modifications to the already designed glyph. The other glyph can be generated without any human interaction.

3.1.3.7 Type 7 analysis

There are 5 glyphs for 2 base glyphs in this type. 3 glyphs, 0DAB.reph, 0DAB0DD2 and 0DAB0DD3 for 0DAB “nna” ණ and 2 glyphs, 0DBD0DD4 and 0DBD0DD4 for 0DBD “la” ළ.

0DAB can be considered as the widest glyph of the glyphs we analyzed. It also contains a diagonal stroke in the middle. This becomes a problem when applying “is-pilla” and “diga is-pilla” and “repaya” for this glyph. We can notice that different fonts have handled this differently. Hence letting the designer decide how these glyphs should be constructed is the best option.

Figure 3.30: How is-pilla is placed in for glyph 0DAB different fonts

0DBD is ending with a descending stroke due to this when applying signs bellow the glyph the bottom part should be changed accordingly so when applying “pa-pilla” and “diga pa-pilla” the bottom part of the glyph should be designed while the top part can be taken from the base glyph 0DBD.

Figure 3.31: How pa-pilla change the glyph 0DBD

For type 7 a total of 14 glyphs should be designed.

3.1.3.8 Type 8 analysis

There are no glyphs for this set.

3.1.3.9 Conclusion for Sinhala Sub Letters

For the 71 glyphs found in Sinhala sub letters 9 new glyphs needed to be designed while another 14 glyphs needed to be designed by doing modifications to the existing glyphs. These statistics are shown in detail in table 3.17.

3.1.4 Numbers

All glyphs for the numbers 0-9 should be designed by the font developer.

Table 3.13: Numbers with Unicode codepoints

0030	0
0031	1
0032	2
0033	3
0034	4
0035	5
0036	6
0037	7
0038	8
0039	9

3.1.5 Punctuations

Since we are not considering extended ASCII characters there are 33 punctuation marks or such symbols in the ASCII character set and majority of these glyphs can be seen in Sinhala legacy fonts.

Table 3.14: Punctuation marks and their availability in legacy fonts

Code points	*glyph*	*description*	*Availability in legacy fonts*
0020		space	100%
0021	!		86.13%
0022	“	Double Quotation mark	12.82%
0023	#		15.12%
0024	$		13.65%
0025	%		96%
0026	&		14.5%
0027	‘	Single Quotation mark	17.65%
0028	(		96.85%
0029	)		99.37%
002A	*		32.14%
002B	+		92.65%
002C	,		100%
002D	-	Hyphen	97.69%
002E	.		99.79%
002F	/		92.85%
003A	:		95.17%
003B	;		81.09%
003C	<		13.24%
003D	=		92.85%
003E	>		13.45%
003F	?		99.37%
0040	@		15.34%
005B	[		73.74%
005C	\		12.61%
005D	]		76.47%

005E	^		10.92%
005F	_	Underscore	14.5%
0060	`	Grave accent	0.84%
007B	{		24.16%
007C	|		17.86%
007D	}		24.37%
007E	~	tilde	0.0%

At this stage it's worth noting that the glyph 0020 which denotes the space is not needed to be designed since it can be generated.

These glyphs share similarities since these glyphs are the inversed glyphs of each other. Hence designing one from these glyphs would be enough. The other glyph can be generated automatically by inversing the designed glyph.

- 0028 "("– 0029 ")"
- 002F "/" – 005C "\"
- 003C "<" – 003E ">"
- 005B "[" – 005D "]"
- 007B "{" – 007D "}"

In most fonts 003A ":" and 003B ";" glyphs are constructed using 002C "." and 002E "," glyphs. Hence, we can come to a conclusion that 003A and 003B are generatable using 002C and 002E. Similarly, 0022 is constructed using two 0027 glyphs.

So, in order to design the 33 punctuation marks the minimum number of glyphs to be designed would be 24 glyphs.

3.1.6 English Letters

The 26 English letters should be designed in both simple and capital variants thus leading to the design of 52 glyphs.

Table 3.15: English letters and their codepoints

0041	A	004E	N	0061	a	006E	n
0042	B	004F	O	0062	b	006F	o
0043	C	0050	P	0063	c	0070	p
0044	D	0051	Q	0064	d	0071	q
0045	E	0052	R	0065	e	0072	r
0046	F	0053	S	0066	f	0073	s
0047	G	0054	T	0067	g	0074	t
0048	H	0055	U	0068	h	0075	u
0049	I	0056	V	0069	i	0076	v
004A	J	0057	W	006A	j	0077	w
004B	K	0058	X	006B	k	0078	x
004C	L	0059	Y	006C	l	0079	y
004D	M	005A	Z	006D	l	007A	z

3.2 Glyphs to be designed for Unicode compliant font

Up until now we were discussing the glyphs to be generated based on the available glyphs in legacy fonts. but there are many glyphs which wasn't discussed in previous sections. This is mainly due to the reason that these glyphs can be generated using the glyphs we discussed previously. But we need to specify which glyphs are going to be included in the font we generate.

Guide to creating Sinhala and Tamil Unicode fonts, contains a chart on how the Sinhala character strings should be placed when sorting them. This is shown in figure 3.32.

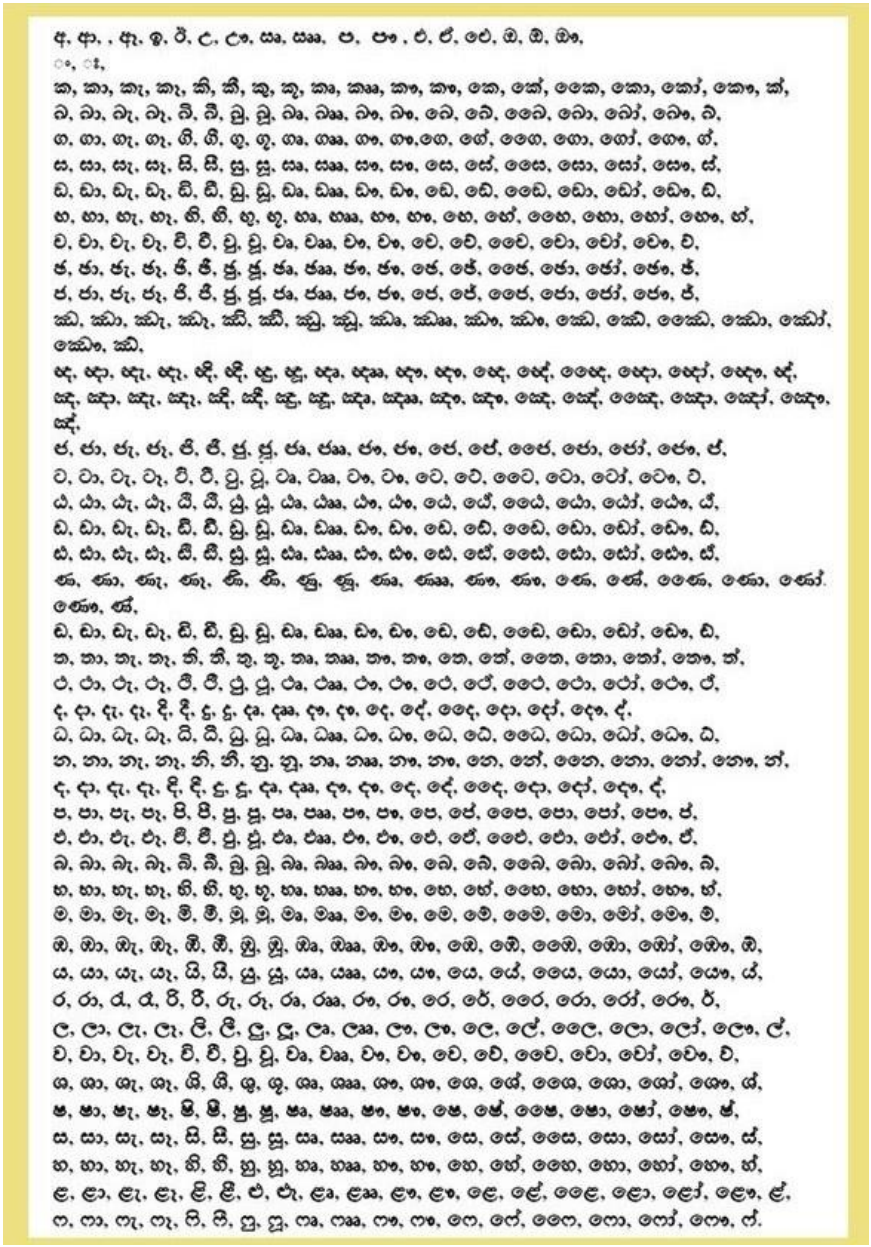

අ, ආ, , ඇ, ඉ, ඊ, උ, ඌ, ඍ, ඎ, ඏ, ඐ, එ, ඒ, ඓ, ඔ, ඕ, ඖ,
ං, ඃ,
ක, කා, කැ, කෑ, කි, කී, කු, කූ, කෘ, කෲ, කෟ, කෳ, කෙ, කේ, කෛ, කො, කෝ, කෞ, ක්,
ඛ, ඛා, ඛැ, ඛෑ, ඛි, ඛී, ඛු, ඛූ, ඛෘ, ඛෲ, ඛෟ, ඛෳ, ඛෙ, ඛේ, ඛෛ, ඛො, ඛෝ, ඛෞ, ඛ්,
ග, ගා, ගැ, ගෑ, ගි, ගී, ගු, ගූ, ගෘ, ගෲ, ගෟ, ගෳ,ගෙ, ගේ, ගෛ, ගො, ගෝ, ගෞ, ග්,
ඝ, ඝා, ඝැ, ඝෑ, ඝි, ඝී, ඝු, ඝූ, ඝෘ, ඝෲ, ඝෟ, ඝෳ, ඝෙ, ඝේ, ඝෛ, ඝො, ඝෝ, ඝෞ, ඝ්,
ඞ, ඞා, ඞැ, ඞෑ, ඞි, ඞී, ඞු, ඞූ, ඞෘ, ඞෲ, ඞෟ, ඞෳ, ඞෙ, ඞේ, ඞෛ, ඞො, ඞෝ, ඞෞ, ඞ්,
ඟ, ඟා, ඟැ, ඟෑ, ඟි, ඟී, ඟු, ඟූ, ඟෘ, ඟෲ, ඟෟ, ඟෳ, ඟෙ, ඟේ, ඟෛ, ඟො, ඟෝ, ඟෞ, ඟ්,
ච, චා, චැ, චෑ, චි, චී, චු, චූ, චෘ, චෲ, චෟ, චෳ, චෙ, චේ, චෛ, චො, චෝ, චෞ, ච්,
ඡ, ඡා, ඡැ, ඡෑ, ඡි, ඡී, ඡු, ඡූ, ඡෘ, ඡෲ, ඡෟ, ඡෳ, ඡෙ, ඡේ, ඡෛ, ඡො, ඡෝ, ඡෞ, ඡ්,
ජ, ජා, ජැ, ජෑ, ජි, ජී, ජු, ජූ, ජෘ, ජෲ, ජෟ, ජෳ, ජෙ, ජේ, ජෛ, ජො, ජෝ, ජෞ, ජ්,
ඣ, ඣා, ඣැ, ඣෑ, ඣි, ඣී, ඣු, ඣූ, ඣෘ, ඣෲ, ඣෟ, ඣෳ, ඣෙ, ඣේ, ඣෛ, ඣො, ඣෝ, ඣෞ, ඣ්,
ඤ, ඤා, ඤැ, ඤෑ, ඤි, ඤී, ඤු, ඤූ, ඤෘ, ඤෲ, ඤෟ, ඤෳ, ඤෙ, ඤේ, ඤෛ, ඤො, ඤෝ, ඤෞ, ඤ්,
ඥ, ඥා, ඥැ, ඥෑ, ඥි, ඥී, ඥු, ඥූ, ඥෘ, ඥෲ, ඥෟ, ඥෳ, ඥෙ, ඥේ, ඥෛ, ඥො, ඥෝ, ඥෞ, ඥ්,
ඦ, ඦා, ඦැ, ඦෑ, ඦි, ඦී, ඦු, ඦූ, ඦෘ, ඦෲ, ඦෟ, ඦෳ, ඦෙ, ඦේ, ඦෛ, ඦො, ඦෝ, ඦෞ, ඦ්,
ඨ, ඨා, ඨැ, ඨෑ, ඨි, ඨී, ඨු, ඨූ, ඨෘ, ඨෲ, ඨෟ, ඨෳ, ඨෙ, ඨේ, ඨෛ, ඨො, ඨෝ, ඨෞ, ඨ්,
ණ, ණා, ණැ, ණෑ, ණි, ණී, ණු, ණූ, ණෘ, ණෲ, ණෟ, ණෳ, ණෙ, ණේ, ණෛ, ණො, ණෝ, ණෞ, ණ්,
ඬ, ඬා, ඬැ, ඬෑ, ඬි, ඬී, ඬු, ඬූ, ඬෘ, ඬෲ, ඬෟ, ඬෳ, ඬෙ, ඬේ, ඬෛ, ඬො, ඬෝ, ඬෞ, ඬ්,
ත, තා, තැ, තෑ, ති, තී, තු, තූ, තෘ, තෲ, තෟ, තෳ, තෙ, තේ, තෛ, තො, තෝ, තෞ, ත්,
ථ, ථා, ථැ, ථෑ, ථි, ථී, ථු, ථූ, ථෘ, ථෲ, ථෟ, ථෳ, ථෙ, ථේ, ථෛ, ථො, ථෝ, ථෞ, ථ්,
ද, දා, දැ, දෑ, දි, දී, දු, දූ, දෘ, දෲ, දෟ, දෳ, දෙ, දේ, දෛ, දො, දෝ, දෞ, ද්,
ධ, ධා, ධැ, ධෑ, ධි, ධී, ධු, ධූ, ධෘ, ධෲ, ධෟ, ධෳ, ධෙ, ධේ, ධෛ, ධො, ධෝ, ධෞ, ධ්,
න, නා, නැ, නෑ, නි, නී, නු, නූ, නෘ, නෲ, නෟ, නෳ, නෙ, නේ, නෛ, නො, නෝ, නෞ, න්,
ඳ, ඳා, ඳැ, ඳෑ, ඳි, ඳී, ඳු, ඳූ, ඳෘ, ඳෲ, ඳෟ, ඳෳ, ඳෙ, ඳේ, ඳෛ, ඳො, ඳෝ, ඳෞ, ඳ්,
ප, පා, පැ, පෑ, පි, පී, පු, පූ, පෘ, පෲ, පෟ, පෳ, පෙ, පේ, පෛ, පො, පෝ, පෞ, ප්,
ඵ, ඵා, ඵැ, ඵෑ, ඵි, ඵී, ඵු, ඵූ, ඵෘ, ඵෲ, ඵෟ, ඵෳ, ඵෙ, ඵේ, ඵෛ, ඵො, ඵෝ, ඵෞ, ඵ්,
බ, බා, බැ, බෑ, බි, බී, බු, බූ, බෘ, බෲ, බෟ, බෳ, බෙ, බේ, බෛ, බො, බෝ, බෞ, බ්,
භ, භා, භැ, භෑ, භි, භී, භු, භූ, භෘ, භෲ, භෟ, භෳ, භෙ, භේ, භෛ, භො, භෝ, භෞ, භ්,
ම, මා, මැ, මෑ, මි, මී, මු, මූ, මෘ, මෲ, මෟ, මෳ, මෙ, මේ, මෛ, මො, මෝ, මෞ, ම්,
ඹ, ඹා, ඹැ, ඹෑ, ඹි, ඹී, ඹු, ඹූ, ඹෘ, ඹෲ, ඹෟ, ඹෳ, ඹෙ, ඹේ, ඹෛ, ඹො, ඹෝ, ඹෞ, ඹ්,
ය, යා, යැ, යෑ, යි, යී, යු, යූ, යෘ, යෲ, යෟ, යෳ, යෙ, යේ, යෛ, යො, යෝ, යෞ, ය්,
ර, රා, රැ, රෑ, රි, රී, රු, රූ, රෘ, රෲ, රෟ, රෳ, රෙ, රේ, රෛ, රො, රෝ, රෞ, ර්,
ල, ලා, ලැ, ලෑ, ලි, ලී, ලු, ලූ, ලෘ, ලෲ, ලෟ, ලෳ, ලෙ, ලේ, ලෛ, ලො, ලෝ, ලෞ, ල්,
ව, වා, වැ, වෑ, වි, වී, වු, වූ, වෘ, වෲ, වෟ, වෳ, වෙ, වේ, වෛ, වො, වෝ, වෞ, ව්,
ශ, ශා, ශැ, ශෑ, ශි, ශී, ශු, ශූ, ශෘ, ශෲ, ශෟ, ශෳ, ශෙ, ශේ, ශෛ, ශො, ශෝ, ශෞ, ශ්,
ෂ, ෂා, ෂැ, ෂෑ, ෂි, ෂී, ෂු, ෂූ, ෂෘ, ෂෲ, ෂෟ, ෂෳ, ෂෙ, ෂේ, ෂෛ, ෂො, ෂෝ, ෂෞ, ෂ්,
ස, සා, සැ, සෑ, සි, සී, සු, සූ, සෘ, සෲ, සෟ, සෳ, සෙ, සේ, සෛ, සො, සෝ, සෞ, ස්,
හ, හා, හැ, හෑ, හි, හී, හු, හූ, හෘ, හෲ, හෟ, හෳ, හෙ, හේ, හෛ, හො, හෝ, හෞ, හ්,
ළ, ළා, ළැ, ළෑ, ළි, ළී, ළු, ළූ, ළෘ, ළෲ, ළෟ, ළෳ, ළෙ, ළේ, ළෛ, ළො, ළෝ, ළෞ, ළ්,
ෆ, ෆා, ෆැ, ෆෑ, ෆි, ෆී, ෆු, ෆූ, ෆෘ, ෆෲ, ෆෟ, ෆෳ, ෆෙ, ෆේ, ෆෛ, ෆො, ෆෝ, ෆෞ, ෆ්,

Figure 3.32: Sinhala collection sequence
Source: (Wijayawardhana and Goonetilleke, 2010)

We can see there are 798 different characters listed in this list. But if we check this with and existing Sinhala Unicode font, we can see that some are not included in the font as glyphs while there are additional glyphs in the font which are not included here.

The signs which are displayed left or right of a glyph is not included in the Unicode fonts. the reason for this is these characters can be displayed using the relevant sign before or after the base glyph. Such signs would be:

ෙ, ෛ, ා, ැ, ෑ, ෘ, ෲ, ෟ, ෳ

Hence the following will not be included as separate glyphs: කැ, කෑ, කෘ, කෲ, කෟ, කෙ, කේ, කෛ, කො, කෝ, කෞ

Those which are included as glyphs but not included in above list are as following:

Bhashitha: කී, කූ, කෟ, ක්‍ර, ක්‍රැ, ක්‍රි, ක්‍රී

Iskola Pota: කී, ක්‍ර, ක්‍රි, ක්‍රී

By analyzing both these fonts we decided the glyphs for each letter to be designed as following

Table 3.16: Glyphs to be designed

	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	10
	! “ # $ % & ‘ ( ) * + , - . / : ; < = > ? @ [\ ] ^ _ ` { | } ~	33
	්, ා , ැ , ෑ , ි, ී, ු, තු, ූ, තූ, ෘ , ෙ, ෟ , ේ, ෛ, ෝ , ෞ	23
	ං, ඃ, අ, ආ, ඇ, ඈ, ඉ, ඊ, උ, ඌ, ඍa, ඍaa, ඏ, ඐ, එ, ඒ, ඓ, ඔ, ඕ, ඖ	20
ක	ක්, කි, කී, කු, කූ, කෘ, ක්‍ර, ක්‍රැ, ක්‍රි, ක්‍රී	11
ඛ	ඛ්, ඛි, ඛී, ඛු, ඛූ, ඛෘ, ඛ්‍ර, ඛ්‍රැ, ඛ්‍රි, ඛ්‍රී	11
ග	ග්, ගි, ගී, ගු, ගූ, ගෘ, ග්‍ර, ග්‍රැ, ග්‍රි, ග්‍රී	11
ඝ	ඝ්, ඝි, ඝී, ඝු, ඝූ, ඝෘ, ඝ්‍ර, ඝ්‍රැ, ඝ්‍රි, ඝ්‍රී	11
ඩ	ඩ්, ඩි, ඩී, ඩු, ඩූ, ඩෘ, ඩ්‍ර, ඩ්‍රැ, ඩ්‍රි, ඩ්‍රී	11
හ	හ්, හි, හී, හු, හූ, හෘ, හ්‍ර, හ්‍රැ, හ්‍රි, හ්‍රී	11
ච	ච්, චි, චී, චු, චූ, චෘ, ච්‍ර, ච්‍රැ, ච්‍රි, ච්‍රී	11
ඡ	ඡ්, ඡි, ඡී, ඡු, ඡූ, ඡෘ, ඡ්‍ර, ඡ්‍රැ, ඡ්‍රි, ඡ්‍රී	11
ජ	ජ්, ජි, ජී, ජු, ජූ, ජෘ, ජ්‍ර, ජ්‍රැ, ජ්‍රි, ජ්‍රී	11

කු	කු, කූ, කෘ, කෲ, කෟ, කෳ, කෲ, කෟ, කෳ	11
කූ	කෑ, කෑ, කෑ, කෑ, කෑ, කෑ, කෑ, කෑ, කෑ	11
ඟූ	ඟෑ, ඟෑ, ඟෑ, ඟෑ, ඟෑ, ඟෑ, ඟෑ, ඟෑ, ඟෑ	11
ඡ	ඡ්, ඡි, ඡී, ඡු, ඡූ, ඡී, ඡ්‍ර, ඡ්‍රි, ඡ්‍රී	11
ට	ට්, ටි, ටී, ටු, ටූ, ටී, ට්‍ර, ට්‍රි, ට්‍රී	11
ඨ	ඨ්, ඨී, ඨී, ඨු, ඨූ, ඨී, ඨ්‍ර, ඨ්‍රි, ඨ්‍රී	11
ඩ	ඩ්, ඩි, ඩී, ඩු, ඩූ, ඩී, ඩ්‍ර, ඩ්‍රි, ඩ්‍රී	11
ඪ	ඪ්, ඪි, ඪී, ඪු, ඪූ, ඪී, ඪ්‍ර, ඪ්‍රි, ඪ්‍රී	11
ණ	ණ්, ණි, ණී, ණු, ණූ, ණී, ණ්‍ර, ණ්‍රි, ණ්‍රී	11
ඬ	ඬ්, ඬි, ඬී, ඬු, ඬූ, ඬී, ඬ්‍ර, ඬ්‍රි, ඬ්‍රී	11
ත	ත්, ති, තී, තු, තූ, තී, ත්‍ර, ත්‍රි, ත්‍රී	11
ථ	ථ්, ථි, ථී, ථු, ථූ, ථී, ථ්‍ර, ථ්‍රි, ථ්‍රී	11
ද	ද්, දා, දැ, දෑ, දි, දී, දු, දූ, දෘ, දෘෘ, දෙෝ, දී, ද්‍ර, ද්‍රා, ද්‍රී, ද්‍රි, ද්‍රී, දෘෘ, දෘෘ	20
ධ	ධ්, ධි, ධී, ධු, ධූ, ධී, ධ්‍ර, ධ්‍රි, ධ්‍රී	11
න	න්, නි, නී, නු, නූ, නී, න්‍ර, න්‍රි, න්‍රී	11
ඳ	ඳ්, ඳා, ඳැ, ඳෑ, ඳි, ඳී, ඳු, ඳූ, ඳී, ඳ්‍ර, ඳ්‍රි, ඳ්‍රී	13
ප	ප්, පි, පී, පු, පූ, පී, ප්‍ර, ප්‍රි, ප්‍රී	11
ඵ	ඵ්, ඵි, ඵී, ඵු, ඵූ, ඵී, ඵ්‍ර, ඵ්‍රි, ඵ්‍රී	11
බ	බ්, බි, බී, බු, බූ, බී, බ්‍ර, බ්‍රි, බ්‍රී	11
භ	භ්, භි, භී, භු, භූ, භී, භ්‍ර, භ්‍රි, භ්‍රී	11
ම	ම්, මි, මී, මු, මූ, මී, ම්‍ර, ම්‍රි, ම්‍රී	11
ඹ	ඹ්, ඹි, ඹී, ඹු, ඹූ, ඹී, ඹ්‍ර, ඹ්‍රි, ඹ්‍රී	11
ය	ය්, යි, යී, යු, යූ, යී	7
ඍ	ඍ්, ඍු, ඍූ, ඍී	5
ර	ර්, රැ, රෑ, රි, රී, රෑ, රෑ	8
ල	ල්, ලි, ලී, ලු, ලූ, ලී, ල්, ලි, ලී	11
ව	ව්, වි, වී, වු, වූ, වී, ව්‍ර, ව්‍රි, ව්‍රී	11
ශ	ශ්, ශි, ශී, ශු, ශූ, ශී, ශ්‍ර, ශ්‍රි, ශ්‍රී	11
ෂ	ෂ්, ෂි, ෂී, ෂු, ෂූ, ෂී, ෂ්‍ර, ෂ්‍රි, ෂ්‍රී	11
ස	ස්, සි, සී, සු, සූ, සී, ස්‍ර, ස්‍රි, ස්‍රී	11
හ	හ්, හි, හී, හු, හූ, හී, හ්‍ර, හ්‍රි, හ්‍රී	11
ළ	ළ්, ළි, ළී	4

ළ	ළෑ	2
ෆ	්, ි, ී, ු, ූ, ෘ, ෟ, ෲ, ෳ, ෟ	11

3.3 Conclusion

This concludes our glyph analysis. In this chapter we identified minimum of 40 glyphs should be generated to create the 123 glyphs in Sinhala base letters and Sinhala sub letters.

Table 3.17: Summary of glyph analysis for base and sub letters

	Sinhala Base Letters			*Sinhala Sub Letters*	*Base + Sub Letters*		
	Glyph count	*Minimum glyphs to be designed*	*Glyphs with changes*	*Glyph count*	*Total*	*Minimum glyphs to be designed*	*Glyphs with changes*
Type 1	18	8	2	43	61	12	2
Type 2	11	5	2	7	18	5	6
Type 3	5	4	-	1	5	4	-
Type 4	2	1	1	10	12	4	4
Type 5	3	3	-	-	3	3	-
Type 6	2	1	1	5	7	1	5
Type 7	9	9	-	5	14	11	3
Type 8	3	-	3	-	3	-	3
Total					123	40	23

For 23 glyphs additional changes should be done but the majority of the glyph can be generated thus saving the designer a huge amount of time.

18 glyphs in sign category are to be designed to generate all 24 glyphs in that set. While all 10 glyphs in number category should be designed. Glyphs for the 33 punctuation marks can be created by designing 24 glyphs. For English letters 52 glyphs are needed to be designed.

When it comes to designing glyphs for Unicode compliant Sinhala fonts the number of glyphs need to be designed increases to 593 glyphs. There's a special character 25CC which is the dotted circle displayed with the signs when they typed without any characters. This should be generated too. For the complete font which would contain 594 glyphs minimum of 144 glyphs

are needed to be designed. and 23 glyphs should be designed by doing modifications to the already designed glyphs. Making a total of 167 glyphs.

Table 3.18: Glyph counts to be designed

	Glyph count	*Minimum glyphs to be designed*	*Glyphs with changes*
English letters	52	52	-
Punctuation marks	33	24	-
Numbers	10	10	-
Signs / Pillam	23	18	-
Sinhala letters	475	40	23
Dotted circle	1	-	-
Total	594	144	23

The developer can also opt to use a font with an open font license such as “LiberationSans” which would make this process quicker. If this approach is taken the need for designing the 86 glyphs under punctuation marks, numbers and English letters would be eliminated. In that case only 81 glyphs needed to be designed including 23 glyphs with additional changes.

3.4 Generate the font using font forge.

With all the glyphs needed to generate the font is identified now we come to the second phase of the research which is generating the font. In this chapter we would design a process of designing a software that would generate the necessary glyphs and then construct the font. We would make the software interactive as possible and giving the font developer instructions to design the glyphs. Since we are designing the software for armature font designers a step-by-step process would be the ideal approach.

In order to create a new font, the developer must provide:

- Font name
- Family name
- Name for humans
- Copyright information
- License information
- Images for the identified minimum glyphs

3.4.1 Glyph Design and Generation Process

The font designers are expected to provide the designed glyphs in png file format with the size of 350 x 350px. The first glyph to be designed would be 0. The design of this glyph serves several purposes. 0 is fairly a simple glyph to be designed. Starting with a simple glyph like that would be easy for the font developer when designing the glyph. Bottom of this glyph should be at line 4 and top of the glyph should be at line 1. Base on the height this glyph we can decide the 5 lines to be followed when designing the future glyphs. The width of this glyph would be taken as the width of the space.

The user will be requested to give a name for the project at the initial stage. All the files that would be created is to be saved in a folder with the given project name. On the next screen the designer would be asked to upload the image created for glyph 0. This glyph will be analyzed and user will be provided with the positions for 5 reference lines to be followed when designing the future glyphs. This would be done by measuring the height of the glyph. Since this glyph is between line 1 and 4 it would mean line 2 is at line 1 + height / 3 and line 3 is at line 1 + height / 2 and line 5 would be at line 4 + height / 3. In the next step the user is required to design the remaining numbers 1 – 9.

If the developer chooses to use LiberationSans font rather than designing Numbers, Punctuation marks and English letters. He would be given 2 options to choose from if the font to be designed has thick lines LiberationSans-Bold would be used otherwise LiberationSans-Regular would be used. The above process will happen here using the 0 glyph of selected LiberationSans font too.

This process can be done in 22 steps as explained in table 3.19. At each step the algorithms will split the uploaded glyphs for necessary parts and reconstruct those parts to construct a new glyph and save the file in the png format. Even though this file format is easy to handled when processing the images when uploading the glyphs vector graphic format is needed.

After all the glyphs are uploaded the png files will be converted to svg files. ImageMagick would be used to convert the png files to pnm file format and Potrace will be used to convert the generated pnm files into svg files.

Table 3.19: Glyph design and generation process

Step	*Glyphs to be designed*	*values*	*Generated glyphs*	*values*
1	0030	0	25CC	◌
2	0031, 0032, 0033, 0034, 0035, 0036, 0037, 0038, 0039	1, 2, 3, 4, 5, 6, 7, 8, 9		
3	0DCA, 0DCF, 0DD0, 0DD1, 0DD2, 0DD3, 0DD4, .0DD4, 0DD6, .0DD6, 0DD8, 0DD9, 0DDF, rakar, reph, yanssaya, 0DF3, 0DF4	්,ා,ැ, ෑ,ි,ී,ු ,ූ,ෘ, ෲ,ෙ, ේ, ෛ, ො, ෝ, ෞ, ෟ	0DF2, 0DDA, 0DDB, 0DDC, 0DDD, 0DDE, yanssaya0DCA, yanssaya0DD4, yanssaya0DD6, yanssaya.reph	ාaa,ේ, ෙෙ,ො, ෞ,ෙෳ, ෂ, ෂූ, ෂූ, ෂ
4	0DBB	ර	0DBB0DCA, 0DBB0DD4, 0DBB0DD6	ර්,රැ,රෑ
	0D8A, 0DBB0DD0, 0DBB0DD1, 0DBB0DD2, 0DBB0DD3	ඊ,ඌ,ඎ,ෆ, ෆ		
5	0D8B, 0D82, 0D83, 0D85, 0D89, 0DAB, 0DBD, 0DC5,	උ,ං,ඃ, අ,ඉ,ණ, ල,ළ,ෆ	0D86, 0D87, 0D88, 0D8C, 0DAB0DCA, 0DAB0DD4, 0DAB0DD6, 0DAB.rakar, 0DAB0DCA.rakar, 0DBD0DCA, 0DBD0DD2, 0DBD0DD3, 0DBD.reph,	ආ,ඇ,ඈ, ඌ, ණ්,ණූ,ණූ, ණූ,ණ්ල්,

	0DC6		0DC50DCA, 0DC50DD2, 0DC50DD3, 0DC60DCA, 0DC60DD2, 0DC60DD3, 0DC60DD4, 0DC60DD6, 0DC6.reph, 0DC6.rakar, 0DC60DCA.rakar, 0DC60DD2.rakar, 0DC60DD3.rakar	ළ්,ළි,ළී,ෆ්, ෆි,ෆී,ෆු,ෆූ, ර්‍ෆ,ෆ්‍ර,ෆ්‍ර්,ෆ්‍රි,ෆ්‍රී
6	0DAB0DD2, 0DAB0DD3, 0DAB.reph, 0DBD0DD4, 0DBD0DD6	ණි,ණී,ර්‍ණ ,ලු,ලූ	0DAB0DD2.rakar, 0DAB0DD3.rakar, 0DBD.rakar, 0DBD0DCA.rakar, 0DBD0DD2.rakar, 0DBD0DD3.rakar	ණ්‍රි,ණ්‍රී,ල්‍ර,ල්‍ර් ,ල්‍රි,ල්‍රී
7	0DA7, 0DAE, 0DA0, 0D94, 0DB90DCA, 0D9E, 0DC00DD2, 0DC00DD3, 0DA80DD2, 0DA80DD3	ට,ථ,ච,ඔ, ඹ්,ඞ,වි,වී, ඨි,ඨී	0D91, 0DB5, 0DB8, 0DC0, 0DA00DCA, 0DA00DD2, 0DA00DD3, 0DA70DCA, 0DA70DD2, 0DA70DD3, 0DAE0DD2, 0DAE0DD3, 0DB50DD2, 0DB50DD3, 0DB80DCA, 0DB80DD2, 0DB80DD3, 0DC00DCA, 0DA8, 0DA9, 0DAA, 0DB0, 0D95, 0D9E0DCA, 0D9E0DD2, 0D9E0DD3, 0DA90DCA, 0DA90DD2, 0DA90DD3, 0DAA0DD2, 0DAA0DD3, 0DB00DCA, 0DB00DD2, 0DB00DD3, 0DB9, 0DB90DD2, 0DB90DD3, 0D9E0DD4, 0D9E0DD6, 0D9E.reph, 0D9E.rakar, 0D9E0DCA.rakar, 0D9E0DD2.rakar, 0D9E0DD3.rakar, 0D92, 0D93, 0D96, 0DA00DD4, 0DA00DD6, 0DA0.reph, 0DA0.rakar, 0DA00DCA.rakar, 0DA00DD2.rakar, 0DA00DD3.rakar, 0DA70DD4, 0DA70DD6, 0DA7.reph, 0DA7.rakar, 0DA70DCA.rakar, 0DA70DD2.rakar, 0DA70DD3.rakar, 0DA80DCA, 0DA80DD4, 0DA80DD6, 0DA8.reph, 0DA8.rakar, 0DA80DCA.rakar, 0DA80DD2.rakar, 0DA80DD3.rakar, 0DA90DD4, 0DA90DD6, 0DA9.reph, 0DA9.rakar, 0DA90DCA.rakar,	එ,ඵ,ම,ව,ච් ,චි,චී,ට්,ටි, ටී,ථි,ථී,ඵි,ඵී ,ම්,මි,මී,ව්, ඨ,ඩ,ඪ,ධ, ඕ,ඞ්,ඞි,ඞී, ඩ්,ඩි,ඩී,ඪි, ඪී,ධ්,ධි,ධී, ඹ,ඹි,ඹී,ඞු, ඞූ,ර්‍ඞ,ඞ්‍ර,ඞ්‍ර්, ඞ්‍රි,ඞ්‍රී,ඒ,ඓ, ඖ,චු,චූ,ර්‍ච, ච්‍ර,ච්‍ර්,ච්‍රි,ච්‍රී,ටු ,ටූ,ර්‍ට,ට්‍ර,ට්‍ර්, ට්‍රි,ට්‍රී,ඨ්,ඨු,ඨූ ,ර්‍ඨ,ඨ්‍ර,ඨ්‍ර්,ඨ්‍රි, ඨ්‍රී,ඩු,ඩූ,ර්‍ඩ, ඩ්‍ර,ඩ්‍ර්,ඩ්‍රි,ඩ්‍රී, ඪ්,ඪු,ඪූ,ර්‍ඪ, ඪ්‍ර,ඪ්‍ර්,ඪ්‍රි,ඪ්‍රී, ථ්,ථු,ථූ,ර්‍ථ, ථ්‍ර,ථ්‍ර්,ථ්‍රි,ථ්‍රී, ධු,ධූ,ර්‍ධ,ධ්‍ර, ධ්‍ර්,ධ්‍රි,ධ්‍රී,ඵ්,ඵු

			0DA90DD2.rakar, 0DA90DD3.rakar, 0DAA0DCA, 0DAA0DD4, 0DAA0DD6, 0DAA.reph, 0DAA.rakar, 0DAA0DCA.rakar, 0DAA0DD2.rakar, 0DAA0DD3.rakar, 0DAE0DCA, 0DAE0DD4, 0DAE0DD6, 0DAE.reph, 0DAE.rakar, 0DAE0DCA.rakar, 0DAE0DD2.rakar, 0DAE0DD3.rakar, 0DB00DD4, 0DB00DD6, 0DB0.reph, 0DB0.rakar, 0DB00DCA.rakar, 0DB00DD2.rakar, 0DB00DD3.rakar, 0DB50DCA, 0DB50DD4, 0DB50DD6, 0DB5.reph, 0DB5.rakar, 0DB50DCA.rakar, 0DB50DD2.rakar, 0DB50DD3.rakar, 0DB80DD4, 0DB80DD6, 0DB8.reph, 0DB8.rakar, 0DB80DCA.rakar, 0DB80DD2.rakar, 0DB80DD3.rakar, 0DB90DD4, 0DB90DD6, 0DB9.reph, 0DB9.rakar, 0DB90DCA.rakar, 0DB90DD2.rakar, 0DB90DD3.rakar, 0DC00DD4, 0DC00DD6, 0DC0.reph, 0DC0.rakar, 0DC00DCA.rakar, 0DC00DD2.rakar, 0DC00DD3.rakar	,ූ,ී,ෟ,ෟ, ෟ,ෟ,ෟ,ෟ, ,ෟ,ෟ,ෟ,ෟ, ෟ,ෟ,ෟ,ෟ, ,ෟ,ෟ,ෟ,ෟ, ෟ
8	0DB6, 0D9B	ඛ,ඛ	0DB60DCA, 0DB60DD2, 0DB60DD3, 0DB60DD4, 0DB60DD6, 0DB6.reph, 0DB6.rakar, 0DB60DCA.rakar, 0DB60DD2.rakar, 0DB60DD3.rakar, 0D9B0DCA, 0D9B0DD2, 0D9B0DD3, 0D9B0DD4, 0D9B0DD6, 0D9B.reph, 0D9B.rakar, 0D9B0DCA.rakar, 0D9B0DD2.rakar, 0D9B0DD3.rakar	ඛ,ඛ,ඛ,ඛ, ඛ,ඛ,ඛ,ඛ, ඛ,ඛ,ඛ,ඛ, ඛ,ඛ,ඛ,ඛ
9	0DAC, 0DC50DD4	ඬ,ඹ	0DAC0DCA, 0DAC0DD2, 0DAC0DD3, 0DAC0DD4, 0DAC0DD6, 0DAC.reph, 0DAC.rakar, 0DAC0DCA.rakar, 0DAC0DD2.rakar, 0DAC0DD3.rakar,	ඬ,ඬ,ඬ,ඬ, ඬ,ඬ,ඬ,ඬ, ඬ,ඬ,ඹ

			0DC50DD6	
10	0D9D, 0DA1, 0DA2, 0DBA, 0DC2	ඝ,ඡ,ජ,ය, ඹ	0DB4, 0DC3, 0D9D0DCA, 0D9D0DD2, 0D9D0DD3, 0D9D0DD4, 0D9D0DD6, 0D9D.reph, 0D9D.rakar, 0D9D0DCA.rakar, 0D9D0DD2.rakar, 0D9D0DD3.rakar, 0DA10DCA, 0DA10DD4, 0DA10DD6, 0DA1.reph, 0DA1.rakar, 0DA10DCA.rakar, 0DA20DCA, 0DA20DD4, 0DA20DD6, 0DA2.reph, 0DA2.rakar, 0DA20DCA.rakar, 0DB40DCA, 0DB40DD2, 0DB40DD3, 0DB40DD4, 0DB40DD6, 0DB4.reph, 0DB4.rakar, 0DB40DCA.rakar, 0DB40DD2.rakar, 0DB40DD3.rakar, 0DBA0DCA, 0DBA0DD2, 0DBA0DD3, 0DBA0DD4, 0DBA0DD6, 0DBA.reph, 0DC20DCA, 0DC20DD2, 0DC20DD3, 0DC20DD4, 0DC20DD6, 0DC2.reph, 0DC2.rakar, 0DC20DCA.rakar, 0DC20DD2.rakar, 0DC20DD3.rakar, 0DC30DCA, 0DC30DD2, 0DC30DD3, 0DC30DD4, 0DC30DD6, 0DC3.reph, 0DC3.rakar, 0DC30DCA.rakar, 0DC30DD2.rakar, 0DC30DD3.rakar, 0D8D, 0D8E	ප,ඇ,ඈ,ඉ,ඊ ,උ,ඌ,ඍ,ඎ, ඏ,ඐ,එ,ඒ,ඓ ,ඔ,ඕ,ඖ,ක, ඛ,ග,ඝ,ඞ,ඟ ,ච,ඡ,ජ,ඣ,ඤ ,ඥ,ඦ,ට,ඨ, ඩ,ඪ,ණ,ඬ,ත ,ථ,ද,ධ,න,ඳ ,ප,ඵ,බ,භ, ම,ඹ,ය,ර,ල ,ව,ශ,ෂ,ස, හ,ළ,ෆ,ා,ැ ,ෑ,ි,ී,ු,ූ, ෘ,ෙ,ේ,ෛ,ො ,ෝ,ෞ,ෟa, ෟaa
11	0DA10DD2, 0DA10DD3, 0DA20DD2, 0DA20DD3, 0DA6, 0D8F	ෂ,ස,හ,ළ, ෆ,෦	0DA10DD2.rakar, 0DA10DD3.rakar, 0DA20DD2.rakar, 0DA20DD3.rakar, 0DA60DCA, 0DA60DD2, 0DA60DD3, 0DA60DD4, 0DA60DD6, 0DA6.reph, 0DA6.rakar, 0DA60DCA.rakar, 0DA60DD2.rakar, 0DA60DD3.rakar, 0D90	ෲ,ෳ,෴,෵,෶ ,෷,෸,෹,෺, ෻,෼,෽,෾,෿ ,෦෧
12	0DC1, 0DC4, 0D9F, 0DB7	ශ,භ,ඖ,ඹ	0D9C, 0D9C0DCA, 0D9C0DD2, 0D9C0DD3, 0D9C0DD4, 0D9C0DD6, 0D9C.reph, 0D9C.rakar, 0D9C0DCA.rakar, 0D9C0DD2.rakar, 0D9C0DD3.rakar,	ග,ග්,ගි,ගී, ගු,ගූ,ගෙ,ගේ, ගෛ,ගො,ගෝ,ග්, ඟි,ඟී,ඟු,ඟූ,

			0D9F0DCA, 0D9F0DD2, 0D9F0DD3, 0D9F0DD4, 0D9F0DD6, 0D9F.reph, 0D9F.rakar, 0D9F0DCA.rakar, 0D9F0DD2.rakar, 0D9F0DD3.rakar, 0DB70DCA, 0DB70DD2, 0DB70DD3, 0DB70DD4, 0DB70DD6, 0DB7.reph, 0DB7.rakar, 0DB70DCA.rakar, 0DB70DD2.rakar, 0DB70DD3.rakar, 0DC10DCA, 0DC10DD2, 0DC10DD3, 0DC10DD4, 0DC10DD6, 0DC1.reph, 0DC1.rakar, 0DC10DCA.rakar, 0DC10DD2.rakar, 0DC10DD3.rakar, 0DC40DCA, 0DC40DD2, 0DC40DD3, 0DC40DD4, 0DC40DD6, 0DC4.reph, 0DC4.rakar, 0DC40DCA.rakar, 0DC40DD2.rakar, 0DC40DD3.rakar	ෙ,ෙ,ෙ,ෙ, ෙ,ෙ,ෙ,ෙ, ෙ,ෙ,ෙ,ෙ, ෙ,ෙ,ෙ,ෙ, ෙ,ෙ,ෙ,ෙ, ෙ,ෙ,ෙ,ෙ, ෙ,ෙ,ෙ
13	0DAF	ද	0DAF0DCA, 0DAF0DCF, 0DAF0DD0, 0DAF0DD1, 0DAF0DD8, 0DAF0DF2, 0DAF0DDD, 0DAF.reph, 0DAF.yanssaya, 0DAF.yanssaya0DCF	ද්,දා,දැ,දෑ, දෘ,දෘෘ,දේ, දී,දෟ,දෟෘ
14	0DB3	ඳ	0DB30DCA, 0DB30DCF, 0DB30DD0, 0DB30DD1, 0DB3.reph	ඳ්,ඳා,ඳැ,ඳෑ, ඳී
15	0DAF.rakar, 0DAF0DD4, 0DAF0DD6, 0DB3.rakar, 0DB30DD4, 0DB30DD6	ෟ,ෟ,ෟ,ෟ,ෟ ,ෟ	0DAF0DD2, 0DAF0DD3, 0DAF0DCF.rakar, 0DAF0DCA.rakar, 0DAF0DD2.rakar, 0DAF0DD3.rakar, 0DB30DD2, 0DB30DD3, 0DB30DD2.rakar, 0DB30DD3.rakar	ි,ී,ෟා,ෟ්,ෟි, ෟී,ෟ,ෟ,ෟ,ෟ
16	0DB1	න		
17	0D9A,0DAD	ක,ත	0DB10DCA, 0DB10DD2, 0DB10DD3, 0DB10DD4, 0DB10DD6, 0DB1.reph, 0DB1.rakar, 0DB10DCA.rakar, 0DB10DD2.rakar, 0DB10DD3.rakar,	න්,නි,නී,නු ,නූ,නෘ,න, න්,ඟි,ඟී,ත්, ති,තී,තු,තූ,

			0DAD0DCA, 0DAD0DD2, 0DAD0DD3, 0DAD0DD4, 0DAD0DD6, 0DAD.reph, 0DAD.rakar, 0DAD0DCA.rakar, 0DAD0DD2.rakar, 0DAD0DD3.rakar, 0D9A0DCA, 0D9A0DD2, 0D9A0DD3, 0D9A0DD4, 0D9A0DD6, 0D9A.reph, 0D9A.rakar, 0D9A0DCA.rakar, 0D9A0DD2.rakar, 0D9A0DD3.rakar	ෆ්,ත,ත්,ත්, ත්,ක්,කි,කී, කු,කූ,කෘ,ක ,ක්,ක්,ක්‍
18	0DA3	ඣ	0DA30DCA, 0DA30DD2, 0DA30DD3, 0DA30DD4, 0DA30DD6, 0DA3.reph, 0DA3.rakar, 0DA30DCA.rakar, 0DA30DD2.rakar, 0DA30DD3.rakar	ඣ,ඣි,ඣී, ඣු,ඣූ,ඣෘ, ඣ්,ඣ්,ඣ්, ඣ්‍
19	0DA4	ඤ	0DA40DCA, 0DA40DD2, 0DA40DD3, 0DA40DD4, 0DA40DD6, 0DA4.reph, 0DA4.rakar, 0DA40DCA.rakar, 0DA40DD2.rakar, 0DA40DD3.rakar	ඤ්,ඤි,ඤී, ඤු,ඤූ,ඤෘ, ඤ්,ඤ්,ඤ්, ඤ්‍
20	0DA5	ඥ	0DA50DCA, 0DA50DD2, 0DA50DD3, 0DA50DD4, 0DA50DD6, 0DA5.reph, 0DA5.rakar, 0DA50DCA.rakar, 0DA50DD2.rakar, 0DA50DD3.rakar	ඥ්,ඥි,ඥී,ඥු ,ඥූ,ඥෘ,ඥ, ඥ්,ඥ්,ඥ්‍
21	0020, 0021, 0023, 0024, 0025, 0026, 0027, 0028, 002A, 002B, 002C, 002D, 002E, 002F, 003C, 003D, 003F, 0040, 005B, 005E, 005F, 0060, 007B, 007C, 007E	!,#,$,%, &,‘,(,*,+ ,,,- ,.,/,<,=,?, @,[,^,_,` ,{,|,~	0022, 0029, 003A, 003B, 003E, 005C, 005D, 007D	“,),:,;,>,\,] ,}
22	0041, 0042, 0043, 0044, 0045, 0046, 0047, 0048, 0049, 004A, 004B,	A, B, C, D, E, F, G, H, I, J, K, L,		

	004C, 004D, 004E, 004F, 0050, 0051, 0052, 0053, 0054, 0055, 0056, 0057, 0058, 0059, 005A, 0061, 0062, 0063, 0064, 0065, 0066, 0067, 0068, 0069, 006A, 006B, 006C, 006D, 006E, 006F, 0070, 0071, 0072, 0073, 0074, 0075, 0076, 0077, 0078, 0079, 007A	M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, a, 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		

The prototype software designed to generate Unicode compliant Sinhala fonts is presented in the following figures.

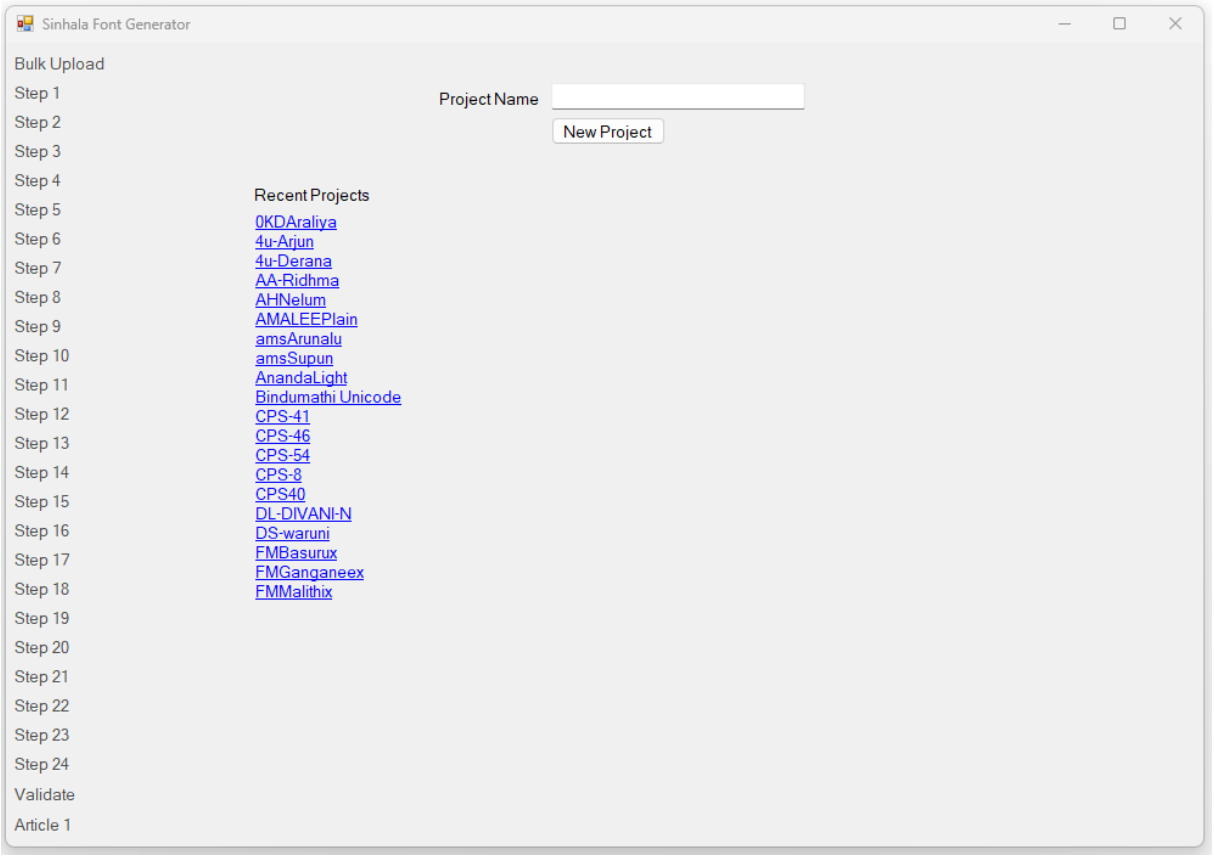


Figure 3.33: Prototype software initial screen

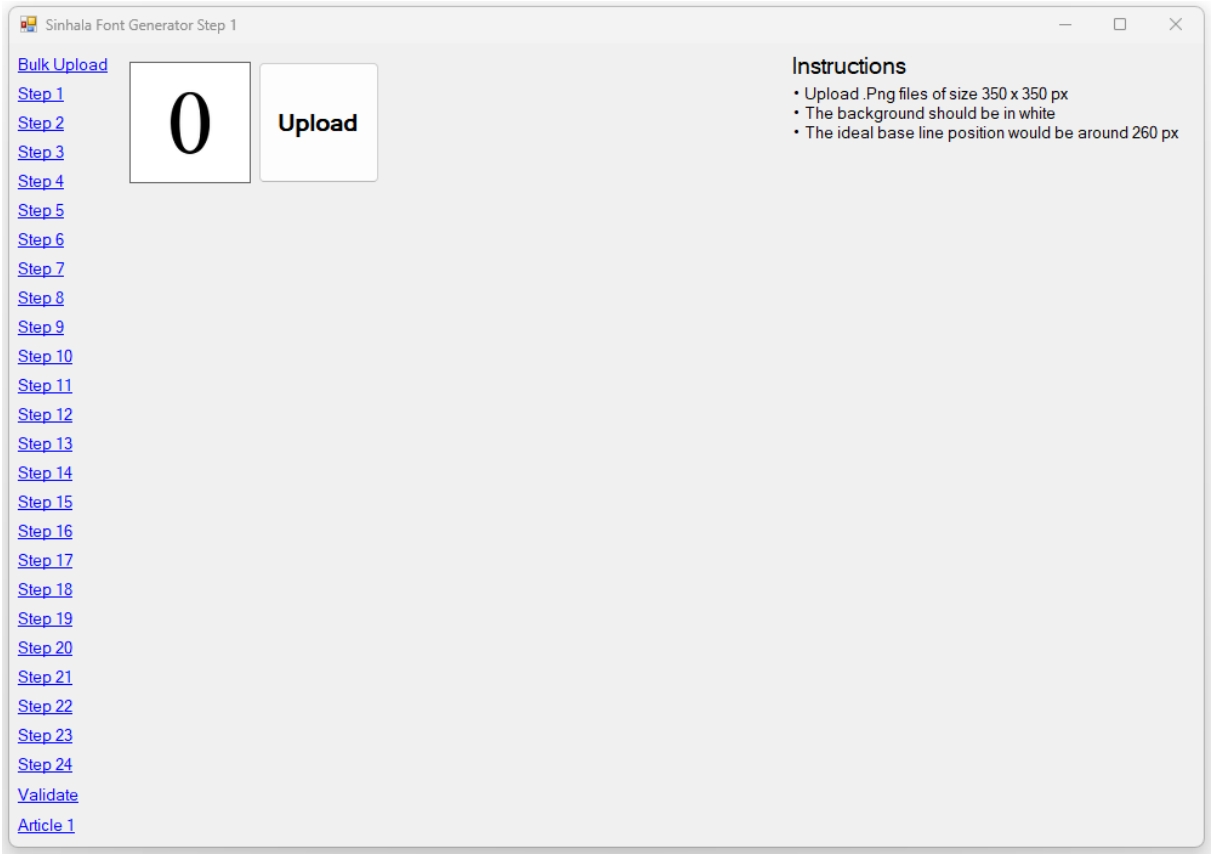


Figure 3.34: Prototype software step 1 – Uploading the first glyph

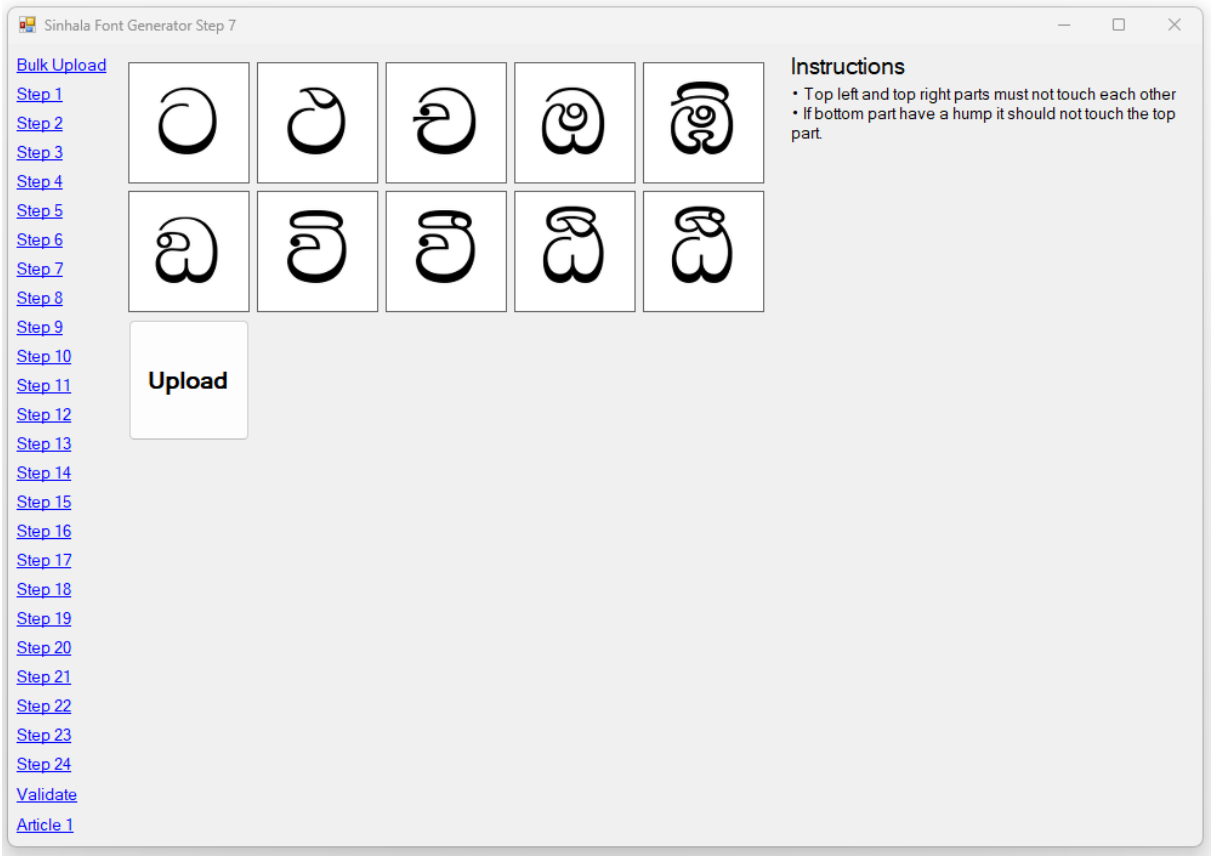


Figure 3.35: Prototype software step 7 – Uploading more Glyphs

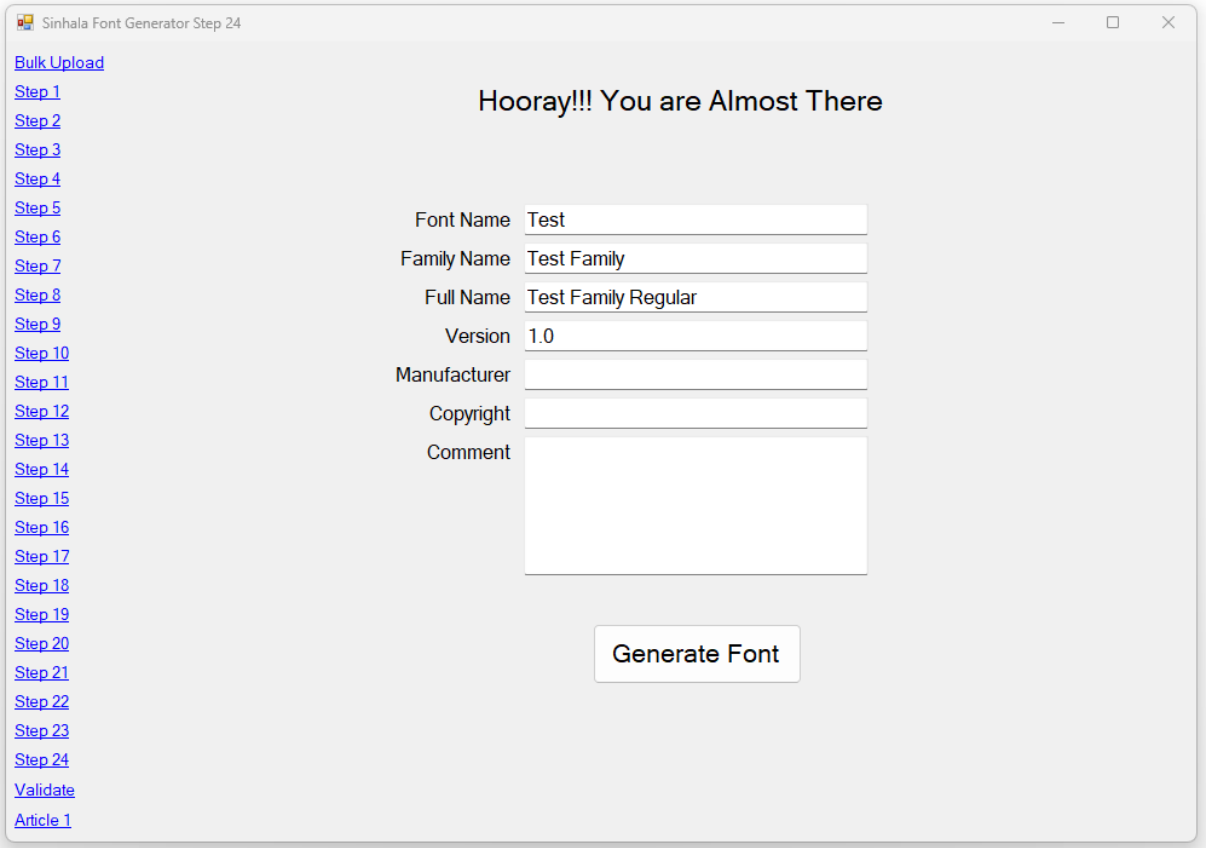


Figure 3.36: Prototype software final step – Generating the font

3.4.2 Font Creation

FontForge allows the creation of fonts in 2 ways. One way is to create a font from scratch while the other way is to use an existing font to modify and create a new font. The following commands can be utilized to create a font:

Table 3.20: FontForge commands for creating fonts.

Command	*Result*
font = fontforge.font()	Create a new font and assign it to the variable “font”
font = fontforge.open(path)	Open a font file given in the “path” and assign it to the variable “font”
glyph = font.createChar(0x0020, 'space')	Create a new glyph with the name “space” in the codepoint 0x0020 and assign it to the “glyph”
font[‘'space’’].clear()	Clear the data of the glyph names “space”. The codepoint 0x0020 can also be used instead of the name. When modifying an

	existing font, this needs to be executed before importing the outlines.
glyph.importOutlines(svg_file_path)	Import the outlines to the glyph using a vector image in the “svg_file_path”
glyphbbox = glyph.boundingBox()	Gets the bounding box information of the glyph.
glyph.correctDirection()	Ensures that the contours in each glyph are oriented consistently
glyph.transform(psMat.translate(x,y))	Changes the position of the glyph
lookup = font.addLookup(“name”, “type”)	Creates a new lookup table with the given name and type
lookup.addLookupSubtable(“name”, “sub_table_name”)	Creates a sub table under lookup table
glyph.addPosSub(“sub_table_name”, positioning_value)	Adds a substitution to the glyph. The positioning_value contains other glyph name and x and y offsets.
font.generate(path)	Generate the new font in a given path with the given name
font.close()	Close the font

Also following attributes needs to be changed when creating a font.

Table 3.21: Attributes to be changed when creating a font with FontForge.

Attribute	*Description*
font.encoding	The encoding used for the font
font.fontname	Name of the font
font.familyname	Family name of the font
font.fullname	Family name and style of the font
font.copyright	Copyright information of the font
font.comment	Additional comments or notes
font.version	The version of the font
font.sfnt_names	Scalar Font Numerical Types
font.em	The em size of the font. Width of capital letter “M”

glyph.width	The width of the glyph

We have utilized these commands and attributes when developing the Unicode compliant Sinhala font generation software. When using an existing font to generate the new font the names that were given to glyphs in the original font should be used correctly when creating glyphs otherwise the substitution rules will be broken.

Once the font is generated the font developer can do the necessary adjustments before publishing the newly created font.

CHAPTER 4
IMPLEMENTATION

In this chapter, the technical details of each step in the designed software will be discussed in detail. The proposed software is a windows forms application. It was developed using C# and .net framework 4.8. FontForge version 20230101 was used for this research. Python scripts were extensively used in the proposed software. All the python scripts are written using python 3.10.0 and the following python packages were used.

- PIL – Image, ImageDraw, ImageFilter
- math
- numpy
- itertools – groupby
- os
- sys
- shutil
- subprocess
- fontforge
- psMat

48 python scripts were used in this software. These scripts are used for several specific tasks. These tasks are listed in Table 4.1 with relevant python scripts and with the step the scripts were used.

Table 4.1: Python scripts used in the software

Task	*Python Script files*	*Usage*
Generating folder structures	GenerateFoldersForGlyphs.py	When a new project is created
Finding split positions	AnalyzeType1.py, AnalyzeType2.py, AnalyzeType3.py, AnalyzeType5.py	Step 1 - Step 20

Splitting glyphs	SplitType1Set1.py, SplitType1Set2.py, SplitType2.py, SplitType3.py, SplitType4.py, SplitType5.py, SplitType8.py	Step 1 - Step 20
Image generation	GlyphGeneration.py	Step 1 - Step 21
Analyzing and image generation for each step	Step1Analyze0Glyph.py, Step2GenerateGlyphs.py, Step3GenerateGlyphs.py, Step4GenerateGlyphs.py, Step5GenerateGlyphs.py, Step6GenerateGlyphs.py, Step7AnalyzeAndSplitType1Set1.py, Step7GenerateGlyphs.py, Step8AnalyzeAndSplitType1Set2.py, Step8GenerateGlyphs.py, Step9AnalyzeAndSplitType1Set2.py, Step9GenerateGlyphs.py, Step10AnalyzeAndSplitType2.py, Step10GenerateGlyphs.py, Step11GenerateGlyphs.py, Step12AnalyzeAndSplitType3.py, Step12GenerateGlyphs.py, Step13GenerateGlyphs.py, Step14GenerateGlyphs.py, Step15GenerateGlyphs.py, Step16AnalyzeAndSplitType5.py, Step16GenerateGlyphs.py, Step17GenerateGlyphs.py, Step18AnalyzeAndSplitType8.py, Step18GenerateGlyphs.py, Step19AnalyzeAndSplitType8.py, Step19GenerateGlyphs.py, Step20AnalyzeAndSplitType8.py, Step20GenerateGlyphs.py,	Step 1 - Step 21

	Step21GenerateGlyphs.py	
Copying files	Step22CopyGlyphsToCommonFolder.py	Step 24
Cropping images	Step23CropGlyphs.py	Step 24
Converting image files	Step24ConvertPngToSvg.py	Step 24
Generating fonts	Step25GenerateFont.py, SinhalaFontGeneration.py	Step 24

When converting a raster image to a vector image file it should be first convert to an intermediate file format Portable Any Map (pnm). This conversion is done using ImageMagick, which is a popular open-source software suite for manipulating images. Once the png files are converted to pnm files, Potrace, an open-source software tool used for tracing bitmap images into vector graphics, is used to convert the pnm files to svg files. Each software provides command line interface to execute their commands and using the commands:

- *magick convert input.png output.pnm*
- *potrace output.pnm -s -o output.svg*

In order to execute python scripts in fontforge the ffpython.exe located in the bin folder of the fontforge installation directory should be utilized. A python script can be executed using ffpython.exe by executing the command:

- *ffpython.exe python_script.py*

SinhalaFontGeneration.py contains the code to create the font. It will be executed using the ffpython.exe and the font will be generated.

BhashithaComplex font was used as the base font for generating the fonts in this research. This was mainly due to the fact that the font can be modified under the Creative Commons Attribution license. So that the glyph substitution rules can be used without reinventing them. It's important to note that not all the glyphs were replaced by the generated glyphs. Hence the better option will be to create a new font which will only contain the necessary glyphs. Thus, reducing the size and complexity of the font.

CHAPTER 5
EVALUTION AND RESULTS

The qualities of a good font can be separated as aesthetics and technical aspects.

A font should be easy to read at various sizes and should facilitate the reading by having well defined letters with proper spacing. And the glyphs should be consistent throughout the font. And should be a visually pleasing for the readers. These are the aesthetic characteristics of a good font.

The technical aspects of a good Unicode compliant font are it should fully support the Unicode standard with proper character encoding. It should contain all the glyphs which is needed to write in that specific language. The font should also support any implementation of open type features such as kerning, glyph substitution etc.

When evaluation the generated font we should consider:

- Readability
- Correctness of character encoding
- Correctness substitution rules

A tool was developed to ensure these evaluation criteria. The generated font can be uploaded to the tool and the user can type in desired text while the text will be displayed on the screen using the uploaded font in multiple sizes. As shown in figure 5.1.

There's also predefined text for all the letters and possible variations of that letters, numbers, signs and even punctuation marks which can be selected by the user. With this the user will be able to easily check each letter and find the issues. Sample of this process is shown in figures 50 and 51

To further clarify if the font complies to the evaluation criteria we have included an paper article ("ව්‍යවස්ථා සභාව අවශ්‍යද?," n.d.) extracted from lankaadeepa.lk. this article contains both Sinhala and English letters and can be viewed using the generated font as shown in figure 52.

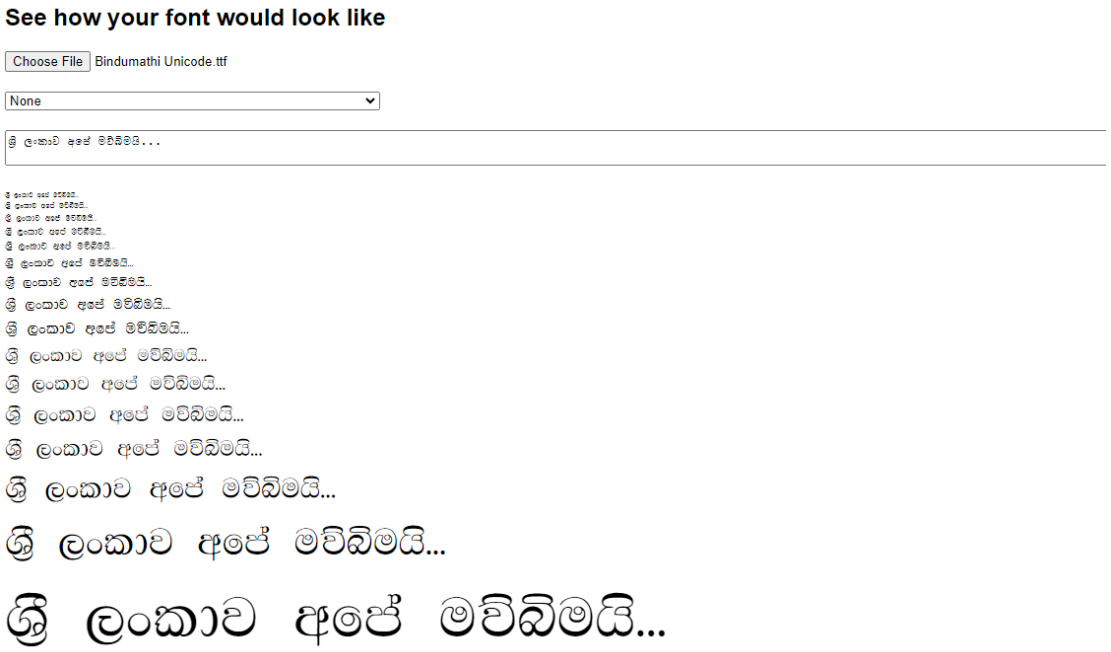


Figure 5.1: Font validation tool with custom text

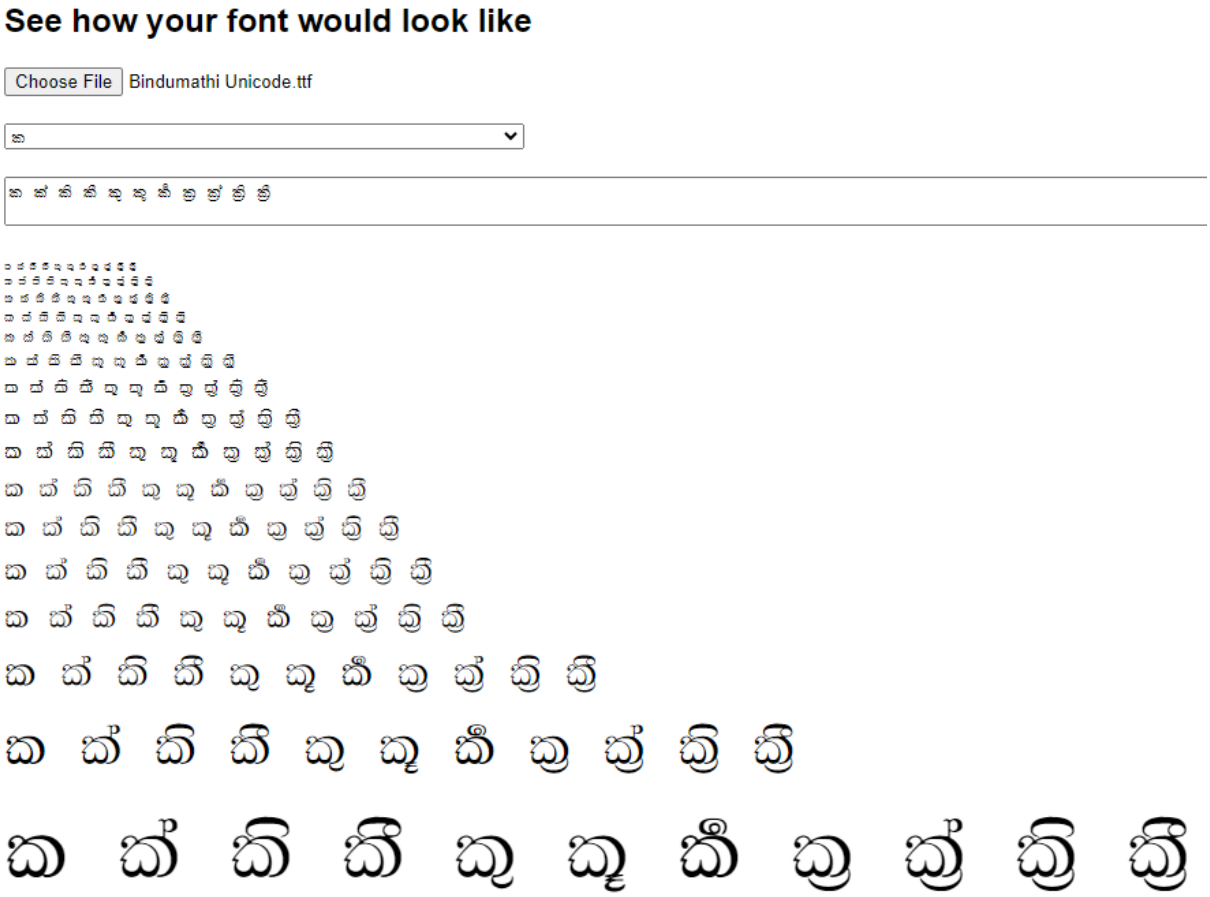


Figure 5.2: Font validation tool with variations of letter “ka”

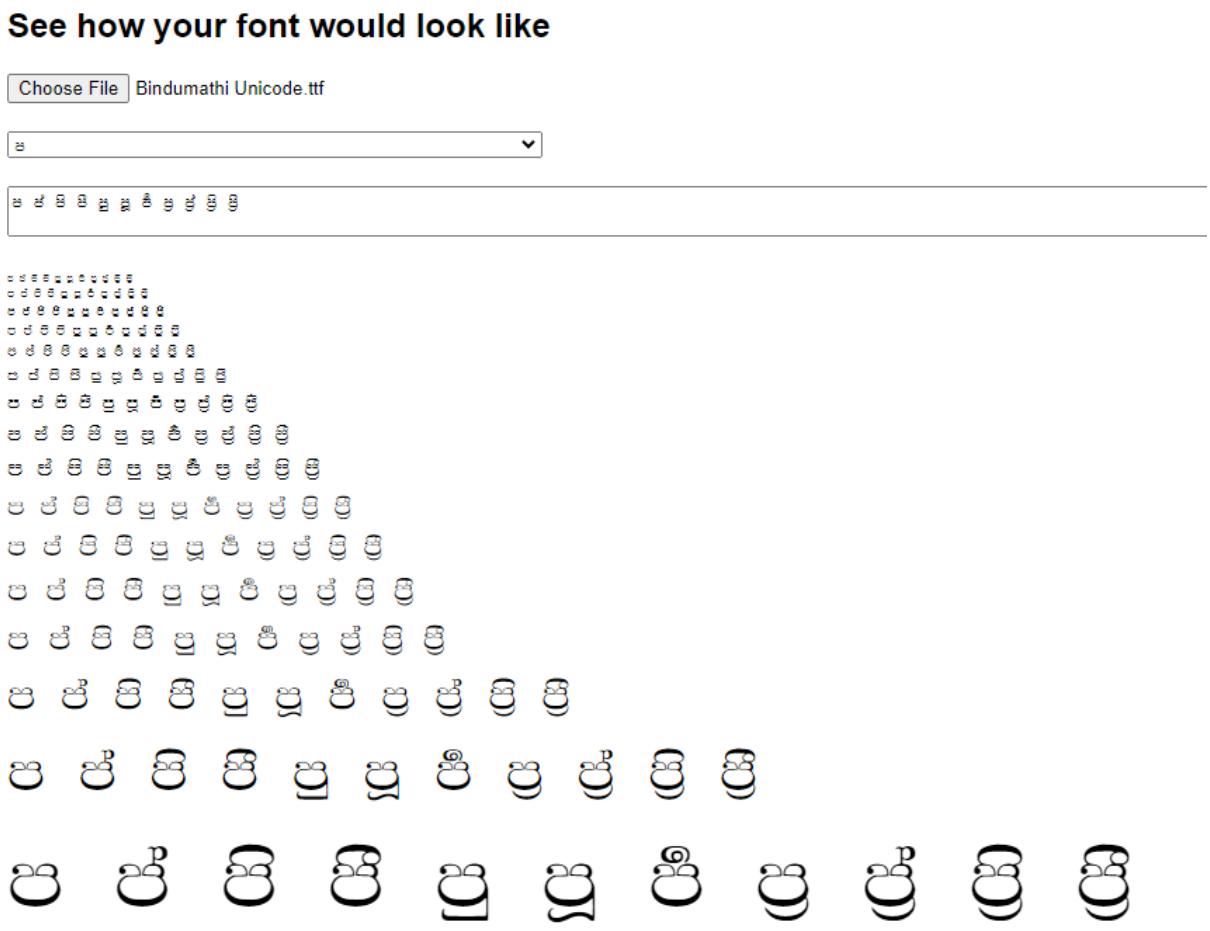


Figure 5.3: Font validation tool with variations of letter “pa”

Figure 5.4: Font validation with paper article

We have used glyphs from the legacy font FM-Bindumathi to create this font. Here while validating the font, we have identified 2 issues of the font.

While majority of the letters in the generated font is readable, the letter “ma” ම and its variants are not properly displayed. This is more noticeable when the size of the font increases.

Figure 5.5: letter “ma” and its variations

When increase the size of the font we can see the joins are not smooth in some letters. Adding a gaussian blur before converting the images to vector format seems to somewhat reduce this effect but still the joining edges can be seen in some letters

Figure 5.6: Non-smooth joints in generated glyphs

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

With the findings of our research, we can conclude that a Unicode compliant Sinhala font can be created by designing 594 glyphs. We identified that the parts of existing glyphs can be used to generate new glyphs. A total of 167 glyphs were identified as the minimum glyphs needed to be designed to create all the 594 glyphs. which consists of,

- 24 Punctuation marks
- 10 Numbers
- 52 English Letters
- 63 Sinhala Letters
- 18 Signs/pillam

This number can be reduced to 81 glyphs by choosing to take glyphs for numbers, English letters and punctuation marks from a font with an open font license. Out of these 81 glyphs, 23 glyphs are designed by doing modifications to existing glyphs.

Once the minimum glyphs are designed, they can be uploaded using a software designed to generate a Unicode compliant Sinhala font using the minimum glyphs. It takes 28 minutes to upload all the 167 glyphs. The time takes to upload the glyphs can be reduced to mere seconds by using the bulk upload option and uploading all the glyphs at once rather than uploading them one by one. User is still needed to go through the steps which will take less than 3 minutes. 1 minute will be taken to fill the meta data and 4 ½ minutes to generate the font.

The process of generating the font once all the minimum glyphs are designed can be done in less than 10 minutes. Implementing this method for Sinhala font creation will significantly hasten the process while enabling anyone with only designing knowledge to create Unicode compliant Sinhala fonts.

6.2 Future works

In this research the suggested prototype software is a windows-based software. But it can be converted into a web-based software enabling more users to create fonts easily without the hassle of installing all the required software on their computers.

An issue with the existing software is the lack of smoothness in the joining points. To address this issue an algorithm is need to be designed to smoothen the joining points of different parts.

Right now, there are a lot of legacy fonts in usage. These fonts contain more than 95% of the glyphs identified as the minimum number of glyphs to be designed. If the original developers of these fonts can take the initiative to convert these legacy fonts to Unicode compliant fonts by using the software developed in this research, a large number of Unicode compliant Sinhala fonts can be created within a short time period.

REFERENCES

Aliyar, 2011. Sinhala Language. Eff. Lang. Learn. URL https://effectivelanguagelearning.com/language-guide/sinhala-language/ (accessed 2.8.24).

Bézier curve, 2024. . Wikipedia.

Brahmi script, 2024. . Wikipedia.

Coperahewa, S., n.d. සිංහල අකුරු පුරාණය. sarasavi publishers.

Disanayaka, J.B., 2006. Sinhala Graphology - සිංහල අක්ෂර විචාරය. Sumitha Publishers.

Disanayaka, J.B., n.d. Approach To Orthography - අක්ෂර වින්‍යාස ප්‍රවේශය. Sumitha Publishers.

Edomoji, 2023. . Wikipedia.

Elu, 2024. . Wikipedia.

Font, 2024. . Wikipedia.

Font editor, 2024. . Wikipedia.

Font hinting, 2024. . Wikipedia.

Glyph, 2024. . Wikipedia.

Kadamba script, 2024. . Wikipedia.

Nandasara, S., K, V., Disanayaka, J., Weerasinghe, R., Wijayawardana, H., 2003. An Introduction to UNICODE for Sinhala Characters. https://doi.org/10.13140/RG.2.1.3654.9604

Nearest-neighbor interpolation, 2023. . Wikipedia.

New Kakuji Font | Webfont & Desktop | MyFonts [WWW Document], n.d. URL https://www.myfonts.com/collections/new-kakuji-font-edomoji-type (accessed 2.9.24).

Prakrit, 2024. . Wikipedia.

Samarawickrama, S., 2017. Anatomy of the Sinhala Letter. J. Eng. Archit. https://doi.org/10.15640/jea.v5n1a9

Sinhala language, 2024. . Wikipedia.

Sinhala script, 2023. . Wikipedia.

Sinhalese language | Sri Lanka, Indo-Aryan, Pali | Britannica [WWW Document], n.d. URL https://www.britannica.com/topic/Sinhalese-language (accessed 2.8.24).

Spatial anti-aliasing, 2024. . Wikipedia.

The Sinhalese language [WWW Document], n.d. URL https://www.17-minute-languages.com/en/blog/learn-more-about-the-sinhalese-language/ (accessed 1.25.24).

The Unicode Standard, Version 15.1, n.d.

TrueType, 2023. . Wikipedia.

Wijayawardhana, H., Goonetilleke, A., 2010. Guide to creating Sinhala and Tamil Unicode fonts.

ව්‍යවස්ථා සභාව අවශ්‍යද? [WWW Document], n.d. URL https://www.lankadeepa.lk/features/ව්‍යවස්ථා-සභාව-අවශ්‍යද-/2-644091 (accessed 2.25.24).

APPENDICES

Analyzed legacy font set:

0KDARALIYA
0KDBOLIDDAbold
0KDMANEL
0KDNAMAL
0KDROSE
0KDSAMAN
4u-Ajanthax
4u-AnandaUltraBold
4u-Anurada
4u-Araliya.Shatter
4u-Arjun
4u-Asirix
4u-Bindumathi
4u-Chami
4u-Derana
4u-Emanee
4u-Ganganee
4u-Gayani
4u-ice
4u-Indumathi
4u-Madu
4u-MANEL
4u-Milith
4u-nisan
4u-Rajantha
4u-Samantha
4u-Sasika
4u-Sawana
4u-SOFT
A-Saman
A-Sarala
A10Yasarath

A12Yasarath
A13Yasarath
AA-Muthu
AA-Ridhma
AA-Shantha
AHNelum
AHQIce
AjithNewS
AKandyNew
AMALEEPlain
AMAmaleeThin
AMAraliyaBold
AMILA-NORMAL
AMParasathuSemiBold
amsArunalu
amsKalana
amsSanduni
amsSevana
amsSevanaBold
amsSupun
amsSwarna
amsTharushi
AnandaHeavy
AnandaLight
AnandaUltraBold
Anuradhabold
AnuradhapuraSupplement
AnuraPlain
Aradana
AradanaBold
AsgiriyaSupplement

ASinhalaAppleMidNormal
BandulaNormal
Br-Ridhma
Bu_Dadunu
Bu_Nilmi
Bu_Nisha
CHAMARA-NORMAL
Chavindra-Classic
CPS-1
CPS-10
CPS-100
CPS-102
CPS-103
CPS-104
CPS-106
CPS-107
CPS-108
CPS-11
CPS-111
CPS-113
CPS-114
CPS-115
CPS-117
CPS-118
CPS-119
CPS-120
CPS-122
CPS-124
CPS-125
CPS-126
CPS-128
CPS-129
CPS-13

CPS-131
CPS-132
CPS-133
CPS-135
CPS-136
CPS-137
CPS-139
CPS-14
CPS-140
CPS-142
CPS-143
CPS-144
CPS-146
CPS-147
CPS-148
CPS-15
CPS-151
CPS-152
CPS-154
CPS-155
CPS-156
CPS-158
CPS-159
CPS-16
CPS-161
CPS-162
CPS-163
CPS-165
CPS-166
CPS-167
CPS-169
CPS-17
CPS-170
CPS-172
CPS-173
CPS-174

CPS-176
CPS-177
CPS-178
CPS-18
CPS-180
CPS-181
CPS-183
CPS-184
CPS-185
CPS-187
CPS-188
CPS-189
CPS-190
CPS-191
CPS-192
CPS-194
CPS-195
CPS-196
CPS-198
CPS-199
CPS-20
CPS-201
CPS-202
CPS-203
CPS-205
CPS-206
CPS-207
CPS-209
CPS-21
CPS-210
CPS-212
CPS-213
CPS-214
CPS-216
CPS-217
CPS-218
CPS-22
CPS-220
CPS-221
CPS-223

CPS-224
CPS-225
CPS-227
CPS-228
CPS-229
CPS-230
CPS-231
CPS-233
CPS-235
CPS-236
CPS-237
CPS-239
CPS-24
CPS-240
CPS-242
CPS-243
CPS-244
CPS-246
CPS-247
CPS-248
CPS-25
CPS-250
CPS-26
CPS-28
CPS-31
CPS-33
CPS-34
CPS-35
CPS-37
CPS-38
CPS-39
CPS-41
CPS-42
CPS-43
CPS-45
CPS-46
CPS-49
CPS-5
CPS-50
CPS-52

CPS-53
CPS-54
CPS-541
CPS-543
CPS-545
CPS-546
CPS-547
CPS-549
CPS-55
CPS-550
CPS-552
CPS-554
CPS-556
CPS-557
CPS-558
CPS-56
CPS-560
CPS-561
CPS-563
CPS-564
CPS-565
CPS-567
CPS-568
CPS-569
CPS-571
CPS-572
CPS-574
CPS-575
CPS-576
CPS-578
CPS-579
CPS-58
CPS-581
CPS-582
CPS-583
CPS-585
CPS-586
CPS-589
CPS-59
CPS-590

CPS-592
CPS-593
CPS-594
CPS-596
CPS-597
CPS-598
CPS-60
CPS-61
CPS-62
CPS-64
CPS-65
CPS-66
CPS-68
CPS-69
CPS-7
CPS-71
CPS-72
CPS-73
CPS-75
CPS-750
CPS-76
CPS-78
CPS-79
CPS-8
CPS-81
CPS-84
CPS-86
CPS-88
CPS-90
CPS-91
CPS-93
CPS-94
CPS-97
CPS-98
CPS-99
CPS40
Devanmini-Future
DharmavathyRegular

Dimuthu
DimuthuBold
DL-Anupama
DL-Anurada
DL-Araliya
DL-Araliya.-841619-
DL-DIVANI-N
Dl-Hansika
Dl-HD
Dl-ice
DL-KIDURU
DL-kumari
DL-Lihini-22
DL-Lihini-ex
DL-Madu.-Cactus
DL-Makela
DL-Malathi
DL-Male
DL-Nelum
DL-PABA
DL-Priyanwada
DL-Pumi
DL-Renu
DL-Ridhma
DL-Ridhma-841619
DL-Ridhma-841619-Mano
DL-Sada
DL-Sarala
DL-Sriyani
DL-Stripe
DL-Sumudu
DS-anuradhiA
Ds-Araliya-AT-Stripe
DS-bursh
DS-Chamika

DS-DAYA-A
DS-malani
Ds-nilantha
DS-waruni
FMAbabldBold
FMAbhayax
FMBasurux
FMEconbldBold
FMGanganeex
FMGemunux
FMMalithix
FMRajanthax
FMRashmeex
FMSandhyaneex
FS-Anupama
FS-Anurada
FS-Araliya
FS-Chami
FS-Dilini-Ltb
FS-Gagani
FS-Javana
FS-Kapila
FS-kumari
FS-Madu.-Cactus
FS-Mali
FS-Manel
FS-MANOlt
FS-Nelum
FS-nisan
FS-Ridhma
FS-Sada
FS-Satsara
FS-Sawana
FS-Sumudu
GS-Anurada
GS-Manori
GS-Nelum
GS-Ridhma
GS-Sumudu

Hela
Helabasa
HW-Anurada
Kandy
KandySupplement
Kapila
Kaputadotcomnormal
KELANIPlain
Kusum
Lankadveepa
Lankanatha
LankapuraSupplement
LankaRuwanSpplement
Lankatilaka
LankatilakaSuppliment
LavanyaRegular
Leonard--S
Leroshon's2ndTTF'97
Leroshon'sFirstTTF'97
Leroshons4TH97
LKMatara98Normal
LMWiithanage-Sinhalese
LmWithaNew1
Mahanuwara
MahanuwaraSupplement
MalithiWeb
ManelNew
ManojafontPlain
Matara

MataraApple1
MataraNormal
MataraSupplement
MENAKA
Mi-Ridhma
Mi_Damindu-Tall
Mi_Damindu-Wide
Mi_Dasun-Tall
Mi_Dasun-Wide
Mi_Dasuni-Wide
Mi_Dasun_96
Mi_Harsha_96
Mi_Nelum2000 Normal
Mi_Nilu2000
Mi_Sarala
NELIKA
NIDAHASAMadu
Niluka-look
NPW-Anju
NPW-Bold
NPW-Chamin
NPW-Deepi
NPW-Sumi
NPW-Tharanga
P-MEDIPlain
Padma
PARASPlain
PiushiNormal
RajeewfontPlain
Ranasuru-PC
Ranaviru-PC
RidhmaboldPlain
RIDHMAPlain

Sandaya-Flintab
Sandaya-light
SARA-BARON
SARA-JANAKEE
SARA-MAHEE
SARA-PRAMESHA
SARA-PUNCHI
SARA-SIRI
SARAWIPULA
Senkadagala
SenkadagalaSupplement
SIN-WalaweBold
SIN-

WalaweBoldItalic
SIN-WalaweNormal
Sinhala--Kelum
Sinhala--KumuduBold
Sinhala--Manel
Sinhala--SupipiSemi-Bold
Sinhala_Bold
SinNelumA
Sisil
SMSChanna
SomiDamayanthi
SomiDilani

SomiDusantha
SomiHansika
SomiNadeeka
SomiNilmi
SomiNilupa
SPlain
Su_ANAGI
Su_ANARGA
Su_ASHA
Su_Eesha
Su_KANTHI
Su_NAGA
Su_NILMINI
Su_NISHA0
Su_Sewandi
Su_SHAKTHI
Su_VIJAYA0

Test--SinhalaMod
Thissamaharama
Thissamaharama Supplement
Tipitaka_Sinhala1
VriSiNhalaCBBold
VriSiNhalaCNNormal
WijesekaraLayout
Wijeya

Glyph availability in legacy fonts as a percentage.

.0DD4	95.798
.0DD6	96.0084
.notdef	99.3697
.notdef2	47.2689
0020	100.4201
0021	86.134
0022	12.8151
0023	15.1260
0024	13.6554
0025	96.0084
0026	14.4957
0027	17.6470
0028	96.8487
0029	99.3697
002A	32.1428
002B	92.6470
002C	100.0
002D	97.6890
002E	99.7899

002F	92.8571
0030	96.8487
0031	96.6386
0032	100.0
0033	100.0
0034	100.0
0035	100.0
0036	100.0
0037	100.0
0038	100.0
0039	100.0
003A	95.1680
003B	81.0924
003C	13.2352
003D	92.8571
003E	13.4453
003F	99.3697
0040	15.3361
005B	73.7394

005C	12.6050
005D	76.4705
005E	10.9243
005F	14.4957
0060	0.8403
007B	24.1596
007C	17.8571
007D	24.3697
00D7	75.2100
00F7	80.6722
0D82	92.6470
0D83	89.9159
0D85	100.0
0D86	0.4201
0D87	0.4201
0D88	0.4201
0D89	100.0
0D8A	99.3697
0D8B	100.0
0D8C	0.8403
0D8D	91.8067
0D8E	0.4201
0D8F	10.5042
0D90	7.9831
0D91	98.9495
0D92	5.6722
0D93	0.4201
0D94	100.0
0D95	96.0084
0D96	0.4201
0D9A	97.2689
0D9A.half	93.6974
0D9A.rakar	2.1008
0D9A.reph	2.5210
0D9A0DCA	2.1008
0D9A0DCA200D0DC2	80.4621
0D9A0DCA200D0DC20DCA	2.5210
0D9A0DCA200D0DC20DD2	2.3109
0D9A0DCA200D0DC20DD3	3.7815
0D9A0DCA200D0DC20DD4	2.5210

0D9A0DCA200D0DC20DD6	0.8403
0D9A0DD2	47.4789
0D9A0DD2.rakar	2.1008
0D9A0DD3	2.9411
0D9A0DD3.rakar	2.1008
0D9A0DD4	2.1008
0D9A0DD6	2.1008
0D9B	99.3697
0D9B.rakar	0.4201
0D9B0DCA	93.067
0D9B0DCA.rakar	0.4201
0D9B0DD2	94.3277
0D9B0DD2.rakar	0.4201
0D9B0DD3	93.2773
0D9B0DD3.rakar	0.4201
0D9B0DD4	2.5210
0D9B0DD6	2.3109
0D9C	99.3697
0D9C.rakar	1.8907
0D9C0DCA	3.5714
0D9C0DCA.rakar	2.5210
0D9C0DD2	1.8907
0D9C0DD2.rakar	2.5210
0D9C0DD3	1.8907
0D9C0DD3.rakar	2.5210
0D9C0DD4	1.8907
0D9C0DD6	1.8907
0D9D	97.8991
0D9D.rakar	2.5210
0D9D0DCA	1.8907
0D9D0DD2	2.5210
0D9D0DD3	2.5210
0D9D0DD4	2.5210
0D9D0DD6	2.5210
0D9E	82.9831
0D9E0DCA	78.5714
0D9E0DD2	60.5042
0D9E0DD2.rakar	0.4201
0D9E0DD3	64.4957
0D9E0DD3.rakar	0.4201

0D9F	95.798
0D9F0DCA	3.3613
0D9F0DD2	3.7815
0D9F0DD3	2.5210
0D9F0DD4	3.5714
0D9F0DD6	2.7310
0DA0	97.2689
0DA0.rakar	0.4201
0DA00DCA	91.5966
0DA00DD2	92.2268
0DA00DD3	91.3865
0DA00DD4	4.2016
0DA00DD6	3.3613
0DA1	98.9495
0DA1.rakar	0.4201
0DA10DCA	89.4957
0DA10DD2	93.2773
0DA10DD2.rakar	0.4201
0DA10DD3	90.5462
0DA10DD3.rakar	0.4201
0DA2	99.5798
0DA2.rakar	2.1008
0DA20DCA	94.7478
0DA20DD2	92.2268
0DA20DD2.rakar	0.4201
0DA20DD3	90.5462
0DA20DD3.rakar	0.4201
0DA20DD4	1.8907
0DA20DD6	1.8907
0DA3	14.0756
0DA3.rakar	0.4201
0DA30DCA	0.8403
0DA30DCA.rakar	0.4201
0DA30DD2	10.5042
0DA30DD2.rakar	0.4201
0DA30DD3	10.2941
0DA30DD3.rakar	0.4201
0DA4	97.8991
0DA4.half	1.0504
0DA40DCA	3.3613

0DA40DCF	56.932
0DA40DD4	72.4789
0DA40DD6	33.1932
0DA5	96.8487
0DA5.half	0.8403
0DA5.rakar	3.7815
0DA50DCF	73.5294
0DA7	97.8991
0DA7.rakar	1.6806
0DA70DCA	96.2184
0DA70DCA.rakar	2.7310
0DA70DD2	93.487
0DA70DD2.rakar	2.7310
0DA70DD3	94.3277
0DA70DD3.rakar	2.7310
0DA70DD4	2.5210
0DA70DD6	2.5210
0DA8	99.1596
0DA8.rakar	0.4201
0DA80DD2	78.3613
0DA80DD2.rakar	0.4201
0DA80DD3	77.5210
0DA80DD3.rakar	0.4201
0DA80DD4	2.5210
0DA80DD6	2.5210
0DA9	97.6890
0DA9.rakar	3.1512
0DA90DCA	93.9075
0DA90DCA.rakar	1.6806
0DA90DD2	93.2773
0DA90DD2.rakar	2.5210
0DA90DD3	92.6470
0DA90DD3.rakar	2.5210
0DA90DD4	1.8907
0DA90DD6	1.6806
0DAA	97.2689
0DAA.rakar	0.4201
0DAA0DD2	75.4201
0DAA0DD2.rakar	0.4201
0DAA0DD3	72.4789

0DAA0DD3.rakar	0.4201
0DAB	100.0
0DAB.reph	68.487
0DAB0DCA	4.4117
0DAB0DD2	29.8319
0DAB0DD3	72.0588
0DAB0DD4	1.8907
0DAB0DD6	1.8907
0DAC	96.2184
0DAC.rakar	0.6302
0DAC0DCA	86.3445
0DAC0DD2	89.4957
0DAC0DD3	88.865
0DAC0DD4	2.9411
0DAC0DD6	2.5210
0DAD	99.3697
0DAD.half	93.487
0DAD.rakar	2.3109
0DAD0DCA	4.2016
0DAD0DCA.rakar	2.3109
0DAD0DCA200D0DAE	72.4789
0DAD0DCA200D0DC0	10.9243
0DAD0DD2	12.6050
0DAD0DD2.rakar	1.8907
0DAD0DD3	2.3109
0DAD0DD3.rakar	1.8907
0DAD0DD4	1.8907
0DAD0DD6	1.8907
0DAE	99.5798
0DAE.rakar	0.4201
0DAE.reph	0.6302
0DAE0DD2	86.3445
0DAE0DD2.rakar	0.4201
0DAE0DD3	86.3445
0DAE0DD3.rakar	0.4201
0DAE0DD4	2.5210
0DAE0DD6	3.5714
0DAF	98.9495
0DAF.half	4.4117
0DAF.rakar	91.5966

0DAF.yansaya	84.4537
0DAF.yansaya0DCF	51.0504
0DAF0DCA	4.2016
0DAF0DCF	72.6890
0DAF0DD0	87.6050
0DAF0DD1	62.6050
0DAF0DD2	93.2773
0DAF0DD2.rakar	5.0420
0DAF0DD3	92.8571
0DAF0DD3.rakar	0.8403
0DAF0DD4	93.2773
0DAF0DD6	93.2773
0DAF0DD8	76.0504
0DAF0DDD	19.9579
0DB0	98.7394
0DB0.rakar	2.7310
0DB00DCA	94.3277
0DB00DCA.rakar	0.4201
0DB00DD2	96.2184
0DB00DD2.rakar	0.4201
0DB00DD3	94.957
0DB00DD3.rakar	0.4201
0DB00DD4	2.5210
0DB00DD6	2.3109
0DB1	97.6890
0DB1.half	93.487
0DB10DCA	2.1008
0DB10DCA200D0DAF	55.2521
0DB10DCA200D0DAF.rakar	0.4201
0DB10DCA200D0DAF.yansaya	6.0924
0DB10DCA200D0DAF.yansaya0 DCF	6.9327
0DB10DCA200D0DAF0DCF	10.9243
0DB10DCA200D0DAF0DD4	0.4201
0DB10DCA200D0DAF0DD6	0.4201
0DB10DD2	14.2857
0DB10DD3	2.3109
0DB10DD4	1.8907
0DB10DD6	1.8907
0DB3	96.0084

0DB3.half	4.2016
0DB3.rakar	1.2605
0DB3.yansaya	73.5294
0DB3.yansaya0DCF	63.0252
0DB30DCA	3.3613
0DB30DCA200D0DB0	8.6134
0DB30DCA200D0DB00DCA	1.2605
0DB30DCA200D0DB00DD2	1.6806
0DB30DCA200D0DB00DD3	0.8403
0DB30DCA200D0DC0	10.0840
0DB30DCA200D0DC00DD2	9.8739
0DB30DCF	61.3445
0DB30DD0	82.7731
0DB30DD1	59.243
0DB30DD2	90.9663
0DB30DD3	91.1764
0DB30DD4	91.3865
0DB30DD6	91.8067
0DB30DDD	58.6134
0DB4	97.6890
0DB4.F	44.9579
0DB4.F0DD2	8.6134
0DB4.F0DD3	3.1512
0DB4.rakar	1.8907
0DB40DCA	4.2016
0DB40DCA.rakar	3.5714
0DB40DD2	3.5714
0DB40DD2.rakar	3.7815
0DB40DD3	3.5714
0DB40DD3.rakar	2.5210
0DB40DD4	1.8907
0DB40DD6	1.8907
0DB5	98.7394
0DB5.rakar	0.4201
0DB50DD2	72.6890
0DB50DD2.rakar	0.4201
0DB50DD3	73.5294
0DB50DD3.rakar	0.4201
0DB50DD4	2.5210
0DB50DD6	2.5210

0DB6	99.3697
0DB6.rakar	2.5210
0DB60DCA	95.5882
0DB60DCA.rakar	2.7310
0DB60DD2	94.3277
0DB60DD2.rakar	2.7310
0DB60DD3	92.8571
0DB60DD3.rakar	1.6806
0DB60DD4	1.8907
0DB60DD6	3.5714
0DB7	98.1092
0DB7.rakar	1.8907
0DB70DCA	2.5210
0DB70DD2	1.8907
0DB70DD3	1.8907
0DB70DD4	1.8907
0DB70DD6	1.8907
0DB8	100.0
0DB8.rakar	0.4201
0DB8.reph	2.9411
0DB80DCA	94.3277
0DB80DCA.rakar	0.4201
0DB80DD2	94.5378
0DB80DD2.rakar	0.4201
0DB80DD3	94.5378
0DB80DD3.rakar	0.4201
0DB80DD4	2.1008
0DB80DD6	1.8907
0DB9	97.4789
0DB90DCA	94.3277
0DB90DD2	95.798
0DB90DD3	94.3277
0DB90DD4	3.7815
0DB90DD6	0.8403
0DBA	100.0
0DBA.reph	77.5210
0DBA0DCA	4.2016
0DBA0DD2	2.3109
0DBA0DD3	2.3109
0DBA0DD4	1.8907

0DBA0DD6	1.8907
0DBB	100.0
0DBB0DCA	90.3361
0DBB0DD0	94.3277
0DBB0DD1	95.5882
0DBB0DD2	93.067
0DBB0DD3	91.3865
0DBB0DD4	1.6806
0DBB0DD6	0.4201
0DBD	100.0
0DBD0DCA	4.2016
0DBD0DD2	7.9831
0DBD0DD3	7.1428
0DBD0DD4	96.8487
0DBD0DD6	96.6386
0DC0	99.7899
0DC0.rakar	0.4201
0DC0.reph	2.5210
0DC00DCA	96.0084
0DC00DCA.rakar	0.4201
0DC00DD2	94.3277
0DC00DD2.rakar	0.4201
0DC00DD3	94.3277
0DC00DD3.rakar	0.4201
0DC00DD4	1.8907
0DC00DD6	1.8907
0DC1	99.5798
0DC1.rakar	3.7815
0DC10DCA	2.3109
0DC10DCA.rakar	2.9411
0DC10DD2	3.3613
0DC10DD2.rakar	3.1512
0DC10DD3	3.3613
0DC10DD3.rakar	55.4621
0DC10DD4	2.9411
0DC10DD6	2.5210
0DC2	97.2689
0DC2.reph	2.5210
0DC20DCA	3.9915
0DC20DD2	2.5210
0DC20DD3	2.5210
0DC20DD4	2.5210
0DC20DD6	2.5210
0DC3	99.1596
0DC30DCA	3.5714
0DC30DD2	1.8907
0DC30DD3	1.8907
0DC30DD4	1.8907
0DC30DD6	3.5714
0DC4	99.5798
0DC40DCA	4.2016
0DC40DD2	1.8907
0DC40DD3	1.8907
0DC40DD4	2.1008
0DC40DD6	1.8907
0DC5	98.3193
0DC50DCA	3.9915
0DC50DD2	3.7815
0DC50DD3	2.5210
0DC50DD4	97.0588
0DC50DD6	0.4201
0DC6	99.5798
0DC6.rakar	1.8907
0DC60DCA	2.3109
0DC60DCA.rakar	1.2605
0DC60DD2	2.5210
0DC60DD2.rakar	1.2605
0DC60DD3	2.5210
0DC60DD3.rakar	1.2605
0DC60DD4	2.5210
0DC60DD6	2.5210
0DCA	95.5882
0DCF	97.8991
0DD0	100.0
0DD1	98.5294
0DD2	95.5882
0DD3	96.0084
0DD4	96.0084
0DD6	95.1680
0DD8	99.5798

0DD9	100.0
0DDA	0.4201
0DDB	0.6302
0DDC	0.4201
0DDD	0.4201
0DDD.sechalf	32.1428
0DDE	0.4201
0DDF	100.0
0DF2	0.6302
0DF3	0.8403
2018	34.6638

2019	69.3277
201C	73.3193
201D	80.2521
half.nasal.stop	40.1260
rakar	94.5378
repaya	94.7478
yansaya	98.1092
yansaya.reph	52.310
yansaya0DD4	2.7310
yansaya0DD6	1.0504