The Evolution of Character Encoding Systems

Daniel Gergov

Period: 7

January 31, 2025

ATCS: Programming Languages

Since the 1960s when the first character encoding standards were released, digital text representation has grown heavily to support increasing technological and linguistic accommodations. The most well-known character encoding standards are ASCII, EBCDIC, and Unicode, each created in different historical contexts and with the goal to solve different issues. ASCII was created in the early 1960s to establish a simple and uniform code to transmit information, while IBM's EBCDIC, developed around the same time, created a unique design shaped by legacy systems. Increasing international technological needs led to the development of Unicode, a standard meant to combine diverse scripts in a single system. This paper will further explore the creation and purpose behind each standard, examining their inner-workings, such as the bit that shifts case in ASCII.

1 American Standard Code for Information Interchange

ASCII is one of the most influential character encoding systems in modern computing, designed to allow the exchange of textual data between various machines and systems. Following an initial plan created by Bob Bemer in 1961, a team was created to work on the new system.[1] Created in 1963 and officially standardized by the American National Standards Institute (ANSI) as X3.4-1963, ASCII was not widely adopted at this time due to the introduction of IBM's EBCDIC. The creation of ASCII was due to an increasing need for a uniform and reliable code that could transmit both textual data and control signals consistently. During a point in time when people used different systems, ASCII was a unifying standard that simplified data exchange.

ASCII is a 7-bit code, therefore able to represent 128 unique characters with numbers from 0 to 127. These characters include two main categories: control codes (positions 0 to 31 and 127) and printable characters (positions 32 to 126). Control codes define actions such as carriage return (CR), line feed (LF), and horizontal tab (HT); therefore, these codes do not print symbols on a screen but instead control text layout and various actions of early technologies such as typewriters. The printable range covers textual symbols, including uppercase and lowercase letters, digits, punctuation marks, and a few special symbols.[2]

One specific and peculiar characteristic of the ASCII design is its organization of letters and symbols. Uppercase Latin letters are given decimal codes 65 to 90, while lowercase Latin letters occupy codes 97 to 122.[3] Ultimately, this organization allows for case manipulation by toggling the sixth bit in the 7-bit ASCII representation; this allows a user of ASCII to quickly shift from uppercase to lowercase and vice versa easily.

As seen in Figure 1, we see the ASCII representations of uppercase and lowercase "a" in binary, decimal, and hexadecimal. In both binary representations, the values are equivalent except for the sixth bit; the sixth bit is 0 for an uppercase Latin letter representation and 1 for a lowercase Latin letter representation.

More specifically, the ASCII system integrated lowercase letters in the next revision of the character encoding system in 1967 with X3.4-1967.[4] Eventually, ASCII became the most popular character encoding system in the United States, but it quickly spread internationally as U.S. computer systems became widespread. Despite ASCII's popularity, the character encoding system's seven-bit limitation meant it could not represent characters outside the basic

[1] Of Encyclopaedia Britannica, 2024.
[2] Awati and Loshin, 2012.
[3] Awati and Loshin, 2012.
[4] Injosoft, n.d.

Character	Binary	Decimal	Hexadecimal
A	0100 0001	65	41
a	0110 0001	97	61

Figure 1: ASCII representation of ”A” and ”a”. Adapted from Awati, R., & Loshin, P. 2012. What is ASCII (American Standard Code for Information Interchange)? https://www.techtarget.com/whatis/definition/ASCII-American-Standard-Code-for-Information-Interchange

Latin alphabet, excluding other languages and symbols. Over time, extensions like “Extended ASCII” fixed these issues by using the eighth bit, but these usually lacked uniformity in the international scene. Therefore, this inherent issue in ASCII in more linguistic coverage led to the creation of separate encoding systems like Unicode.[5]

2 Extended Binary Coded Decimal Interchange Code

EBCDIC is a character encoding standard created by IBM in 1963 as part of its System/360 and System/370 mainframe architecture. EBCDIC’s roots trace earlier than ASCII to punch card systems and earlier encoding schemes, specifically IBM’s Binary Coded Decimal (BCD) encoding. The motivation was to allow for the backward compatibility for large amount of data and programs already stored on IBM equipment; ultimately, this allows users to easily transition to the new and smoother System/360 line.

EBCDIC is an 8-bit code, allowing for 256 possible unique values. However, the organization of characters in the EBCDIC system differs from ASCII in the fact that EBCDIC’s encodings are not organized sequentially, but rather on usage frequency. For example, Latin alphabetic characters may be split across different code ranges, which may make programming and communication more complex in tasks such as sorting.[6] This non-sequential organization originates from older punch-card systems, where certain columns and rows were reserved for other purposes. Therefore, since users relied on these older systems, IBM ensured that the transition between older systems and EBCDIC mainframes was more straightforward for customers, but it also introduced challenges for maintaining compatibility with other platforms.

Due to IBM’s dominance in the international mainframe market, EBCDIC remained a standard in computing for decades. Since the introduction of the EBCDIC, various extensions called code pagess have been added to the encoding standard. For example, the addition of code page 930 (CCSID 930) added the support to encode various Japanese characters. However, these extensions to EBCDIC did not gain much popularity due to poor design choices such as the reliance on Shift-In and Shift-Out codes, making it difficult to parse a byte sequence from the middle without backtracking.[7] Unlike ASCII which grew into a universal standard, EBCDIC remained confined to IBM’s ecosystem, preventing its further adoption.

[5] Awati and Loshin, 2012.
[6] 60sec.site, 2025.
[7] Contributors, 2023.

3 Unicode

Following the realization that ASCII and other encoding systems were insufficient for an international character encoding, Joe Becker, Lee Collins, and Mark Davis studied the potential of a new global encoding system.[8] The Unicode Consortium, a group of industry giants including Apple, Xerox, and Microsoft, then released Unicode in January of 1991. The standard attempted to combine all international writing systems under one character set. The motivation behind Unicode was the need to ensure proper communication across various linguistic boundaries, allowing text data to be uniformly represented across all platforms.[9]

In Unicode's architecture, every character is assigned a "code point" which is a hexadecimal value. The large amount of possible characters is divided into various "planes," with the most commonly used scripts (Latin, Cyrillic, and Arabic) housed in the Basic Multilingual Plane (BMP). Unicode also contains other planes that contain rarely-used scripts, emojis, and special symbols. Ultimately, this highly organized system of planes allows Unicode to expand as needed once new character sets are to be supported.[10]

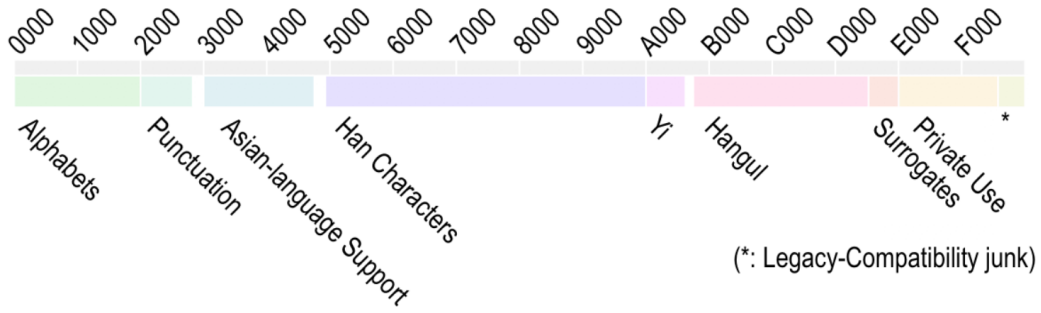

Figure 2: A diagram showing the organization of the Basic Multilingual Plane. Adapted from Bray, T. (2006, October 22). Basic Multilingual Pane. Wikimedia Commons. https://commons.wikimedia.org/wiki/File:Basic_Multilingual_Pane.png.

As seen in Figure 2, the organization of the BMP is divided into sequential blocks of code points, each dedicated to specific scripts or categories. It spans from U+0000 to U+FFFF, containing the most commonly used characters in current writing systems.

Another characteristic of Unicode is its ability to essentially combine visually and semantically identical symbols that are contained in different regional encodings. For example, a character with the same shape and meaning across languages may be mapped to the same code point, ultimately reducing confusion of having different code points for analogous symbols.[11]

The Unicode standard also specifies several encoding forms, such as UTF-8, UTF-16, and UTF-32, each created for different use cases. The Unicode Transition Format (UTF) defines these encoding forms, where the UTF-8 character set defines the storage of code points into 8 bits. UTF-8 is the most popular character set on the internet, used within nearly every webpage. Moreover, UTF-8 remains backward-compatible with ASCII for the 7-bit range, only expanding to additional bytes for characters outside the BMP. This efficiency has led UTF-8 to become the standard format for online text, allowing the use of multilingual data in many

[8] Contributors, 2025.
[9] Contributors, 2025.
[10] Contributors, 2025.
[11] Contributors, 2025.

websites and databases.[12]

4 Conclusion

Due to its flexible design and global support, Unicode has become the most popular encoding system in modern text processing. The Unicode Consortium continues to update the standard with new versions that add new symbols and emojis, ensuring that Unicode represents every character. Through these updates, Unicode acts as the global character encoding standard, accomplishing the goals of ASCII, EBCDIC, and other encoding systems.[13]

References

60sec.site. (2025). What is ebcdic in computing? *60sec.site*. https://60sec.site/terms/what-is-ebcdic-in-computing-extended-binary-coded-decimal-interchange-code

Awati, R., & Loshin, P. (2012). What is ascii (american standard code for information interchange)? *TechTarget*. https://www.techtarget.com/whatis/definition/ASCII-American-Standard-Code-for-Information-Interchange

Consortium, T. U. (1991). History of unicode release and publication dates. *unicode.org*. https://www.unicode.org/history/publicationdates.html

Contributors, W. (2023). Character encodings/code tables/ebcdic/code page 930. *Wikibooks*. https://en.wikibooks.org/wiki/Character_Encodings/Code_Tables/EBCDIC/Code_page_930

Contributors, W. (2025). Unicode. *Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/wiki/Unicode

Injosoft. (n.d.). The history of ascii 1963. *Injosoft*. https://www.asciicode.com/articles/ASCII1963#:~:text=ASCII%201963%2C%20also%20known%20as,IBM's%20proprietary%20character%20set%2C%20EBCDI.

of Encyclopaedia Britannica, T. E. (2024). Ascii. *Encyclopaedia Britannica*. https://www.britannica.com/topic/ASCII

Soubiran, T. (2020). Utf : Unicode transformation format. *NUMA (OpenEdition)*. https://numa.hypotheses.org/626

[12] Soubiran, 2020.

[13] Consortium, 1991.