Non-Numerical Coding Systems

Everything has a Binary Representation

Inside the computer all values are represented in ones and zeros. In the previous lessons we saw how numbers, even negative numbers can be represented by a string of ones and zeros. However even letters of the alphabet have a corresponding binary value. Even the boolean values. Even operations (e.g. add, subtract etc.) have their code. All data inside the computer is made of ones and zeros.

Looking at some Codes

ASCII

- **ASCII** is the acronym for the American Standard Code for Information Interchange.
- It is a code for representing 128 English characters as numbers, with each letter assigned a number from 0 to 127.
  - For example, the ASCII code for uppercase M is 1001101_2 i.e. $4D_{16}$ or 77_{10}.
  - Each ASCII code consists of 7 bits.
  - Note that $2^7 = 128$.
- Most computers use ASCII codes to represent text.
- Text files (a **text file** contains only text, no numbers, no boolean values etc) stored in ASCII format are also called **ASCII files**).
- Below is a picture showing a part of the ASCII code.

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
32	20	Space	64	40	@	96	60	`
33	21	!	65	41	A	97	61	a
34	22	"	66	42	B	98	62	b
35	23	#	67	43	C	99	63	c
36	24	$	68	44	D	100	64	d
37	25	%	69	45	E	101	65	e
38	26	&	70	46	F	102	66	f

- The picture above shows a section of the printable characters but there are also non-printable characters e.g. 'delete' (ASCII code 127_{10}).

- **Extended ASCII** reserves 8 bits for each character. This takes the total number of different characters from 128 to 256. A part of the Extended ASCII code is shown in the following picture.

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
128	80	Ç	160	A0	á	192	C0	└	224	E0	α
129	81	ü	161	A1	í	193	C1	┴	225	E1	ß
130	82	é	162	A2	ó	194	C2	┬	226	E2	Γ

EBCDIC

- **EBCDIC** is the acronym for Extended Binary Coded Decimal Interchange Code.
- It is a binary code for alphabetic and numeric characters that IBM developed (in 1964 and 1964) for its larger computers.
- Each alphabetic or numeric character is represented with an 8-bit binary number. 256 possible characters (letters of the alphabet, numerals, and special characters) are defined.
- IBM's PCs and workstations do not use EBCDIC. Instead, they use the industry standard code for text, ASCII.
- Conversion programs allow different operating systems to change a file from one code to another.

Unicode

- **Unicode** provides a unique number for every character (as do all character codes).
- It covers letters and symbols from all the languages of the world.
- It has been adopted by all modern software providers and now has become a standard.
- Thus data can be transported through many different platforms, devices and applications without the need for translation.
- The Unicode Consortium is a non-profit organization founded to develop, extend and promote use of the Unicode Standard.
- The development of Unicode is ongoing. Version 13.0 of the Unicode Standard contains 143,859 characters

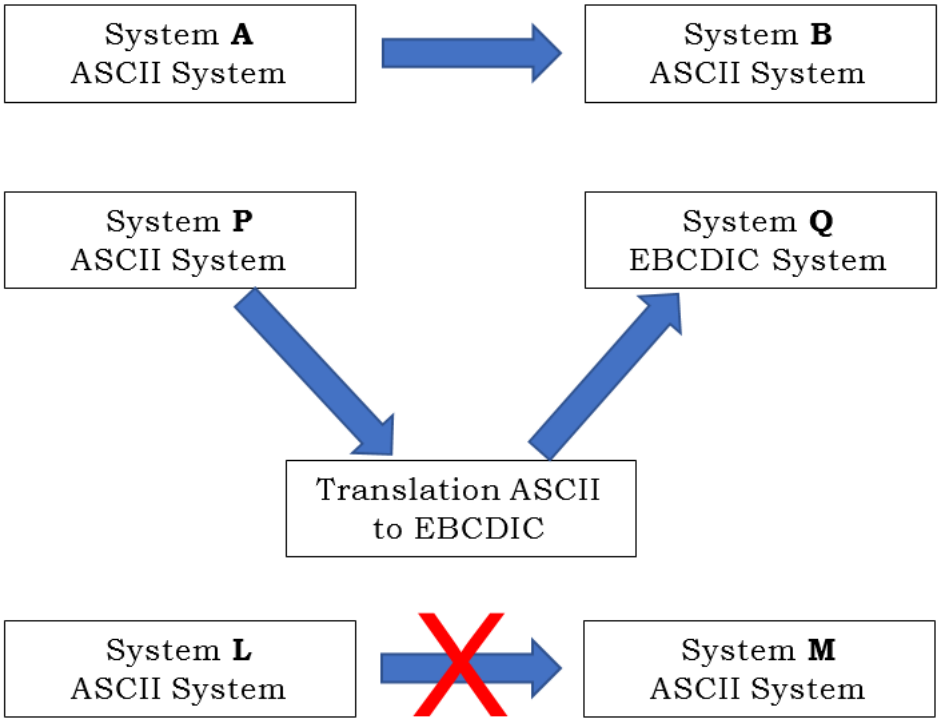
Portability

Data portability is the capacity to move data from one place to another in such a way that if data is sent from system X to system Y, then system Y has ways and means of understanding the data.

In the following diagram, system A is compatible with system B. They both read and understand ASCII. So, there is portability.

Systems P and Q are not compatible (one system is ASCII and the other is EBCDIC). However, system Q is equipped with a translator from ASCII to EBCDIC. In this case there is portability.

In the absence of ASCII / EBCDIC translators, systems L and M cannot receive each other's data as they have no means of interpreting the content. In this case there is no data portability between systems L and M.



Some Considerations

	ASCII		EBCDIC		Unicode	
A	65	1000001	193	11000001	65	1000001
B	66	1000010	194	11000010	66	1000010
C	67	1000011	195	11000011	67	1000011

The codes in the table above are given in both decimal and binary. Note that the codes of subsequent letters have subsequent numbers. If A is 65_{10} then B is 66_{10} etc.

The table below shows the codes for the characters ‘a’, ‘b’ and ‘c’.

	ASCII		EBCDIC		Unicode	
a	97	1100001	129	10000001	97	1100001
b	98	1100010	130	10000010	98	1100010
c	99	1100011	131	10000011	99	1100011

Note that the codes are sequential. Note also that the difference between a lowercase letter and an uppercase letter is 32_{10} for the ASCII and Unicode codes. This amounts to the sixth bit from the right. In the case of EBCDIC the difference is 64_{10} so in the binary code the only difference between a lowercase and an uppercase letter is the seventh bit from the right.

The codes are worked out to facilitate text processing for example:

1. Sorting is easier because the codes are made in such a way that if ‘m’ is smaller (in alphabetical order) than ‘r’, then the code of ‘m’ is also smaller than the code of ‘r’.
2. Changing letters from lowercase to uppercase would amount to changing the value of just one bit.