CM1604
Computer Systems Fundamentals

Data Representation
Computing Components

Week No 03| Lecturer Name

In this week lecture..

- Data storage
- Character encoding
  - ASCII
  - UNICODE
- Images
  - 24 bit colour RGB
  - Sizes of images
- Computing components

By the end of this lecture, you will be able to:

- Distinguish between various units of data storage and convert between them
- Understand standards of textual representation for computer systems
  - ASCII
  - UNICODE
- Understand 24-bit RGB colour coding and calculate the size of bitmap images
- List the components and their function in a von Neumann machine.

Data Storage

Bit

- Binary Digit
- 0 or 1 → smallest unit

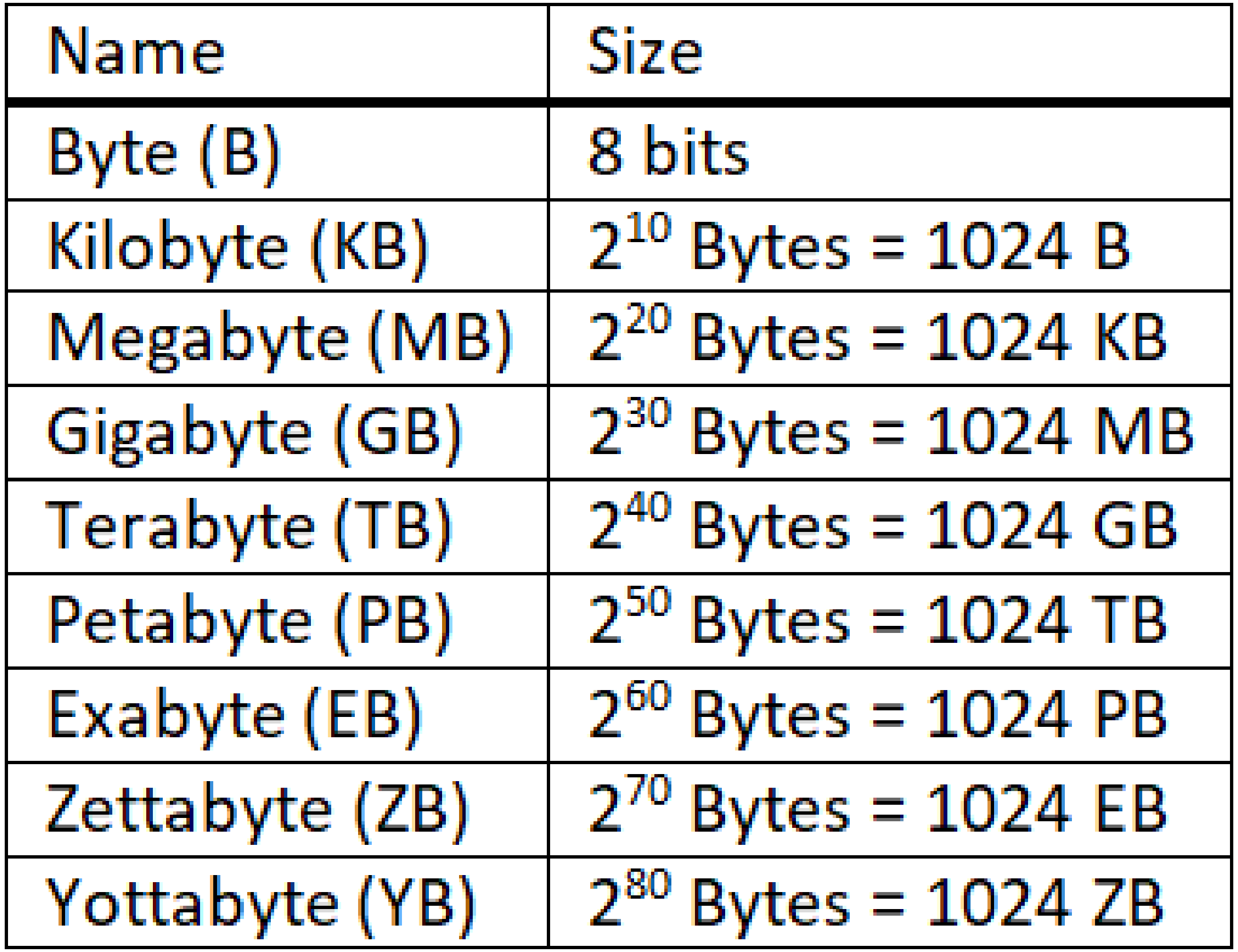
Nibble

- 4 bits → 1 Hex

Byte

- 8 bits
- Smallest addressable unit in computer

Data Storage



Name	Size
Byte (B)	8 bits
Kilobyte (KB)	2^{10} Bytes = 1024 B
Megabyte (MB)	2^{20} Bytes = 1024 KB
Gigabyte (GB)	2^{30} Bytes = 1024 MB
Terabyte (TB)	2^{40} Bytes = 1024 GB
Petabyte (PB)	2^{50} Bytes = 1024 TB
Exabyte (EB)	2^{60} Bytes = 1024 PB
Zettabyte (ZB)	2^{70} Bytes = 1024 EB
Yottabyte (YB)	2^{80} Bytes = 1024 ZB

Character encoding

Representing text

- How to represent text?
  - Number of characters → finite
  - Then assign each character a binary value /pattern

- Character set
  - A list of character and the codes used to represent them
  - Standard agreed by the manufacturers

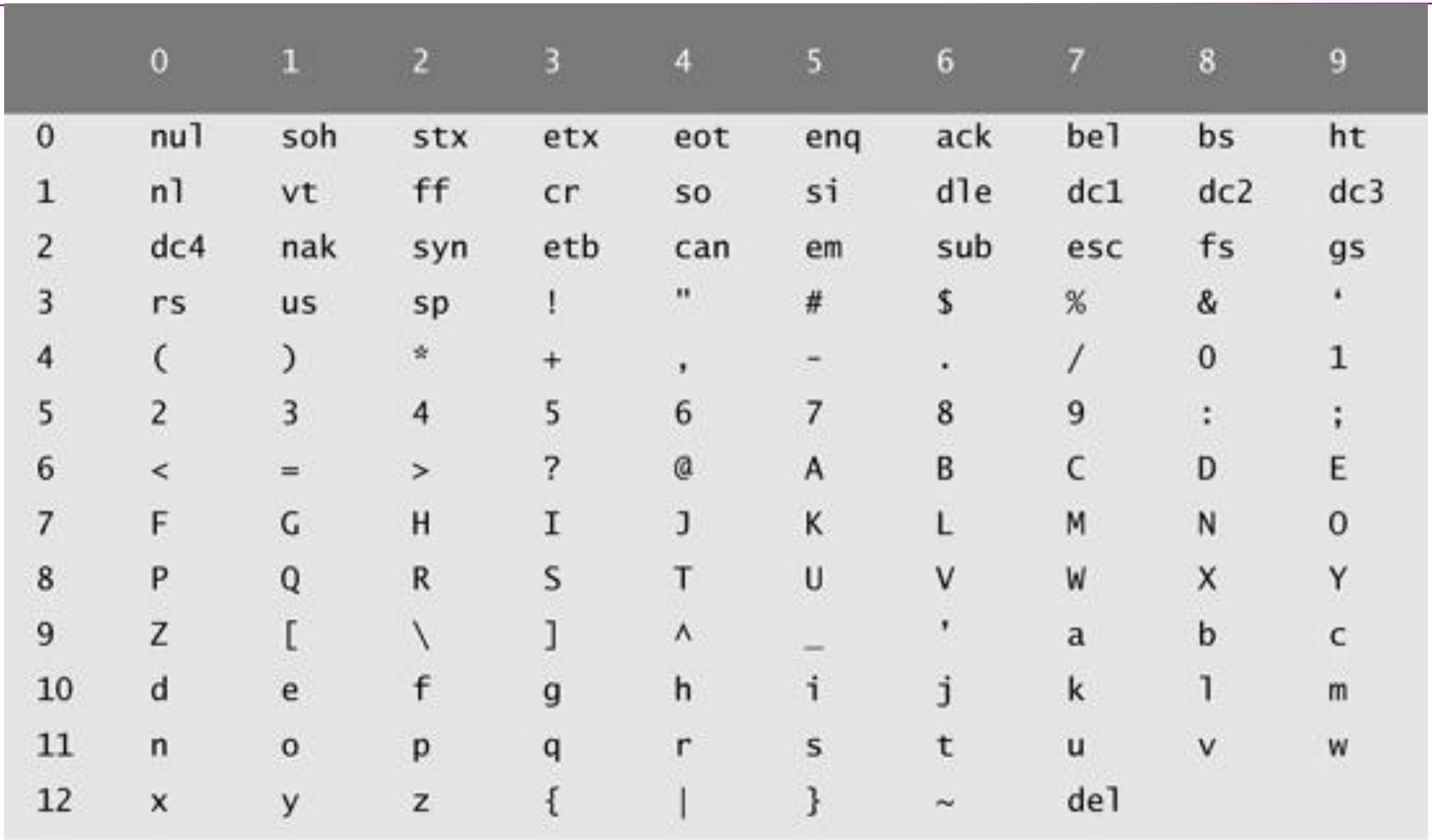
  - ASCII
  - Unicode

ASCII Character set

- **A**merican **S**tandard **C**ode for **I**nformation **I**nterchange
- Initially used 7-bits → 128 unique characters

- Evolved version → Extended ASCII
  - Use 8-bits
  - 256 possible unique characters

ASCII Character set



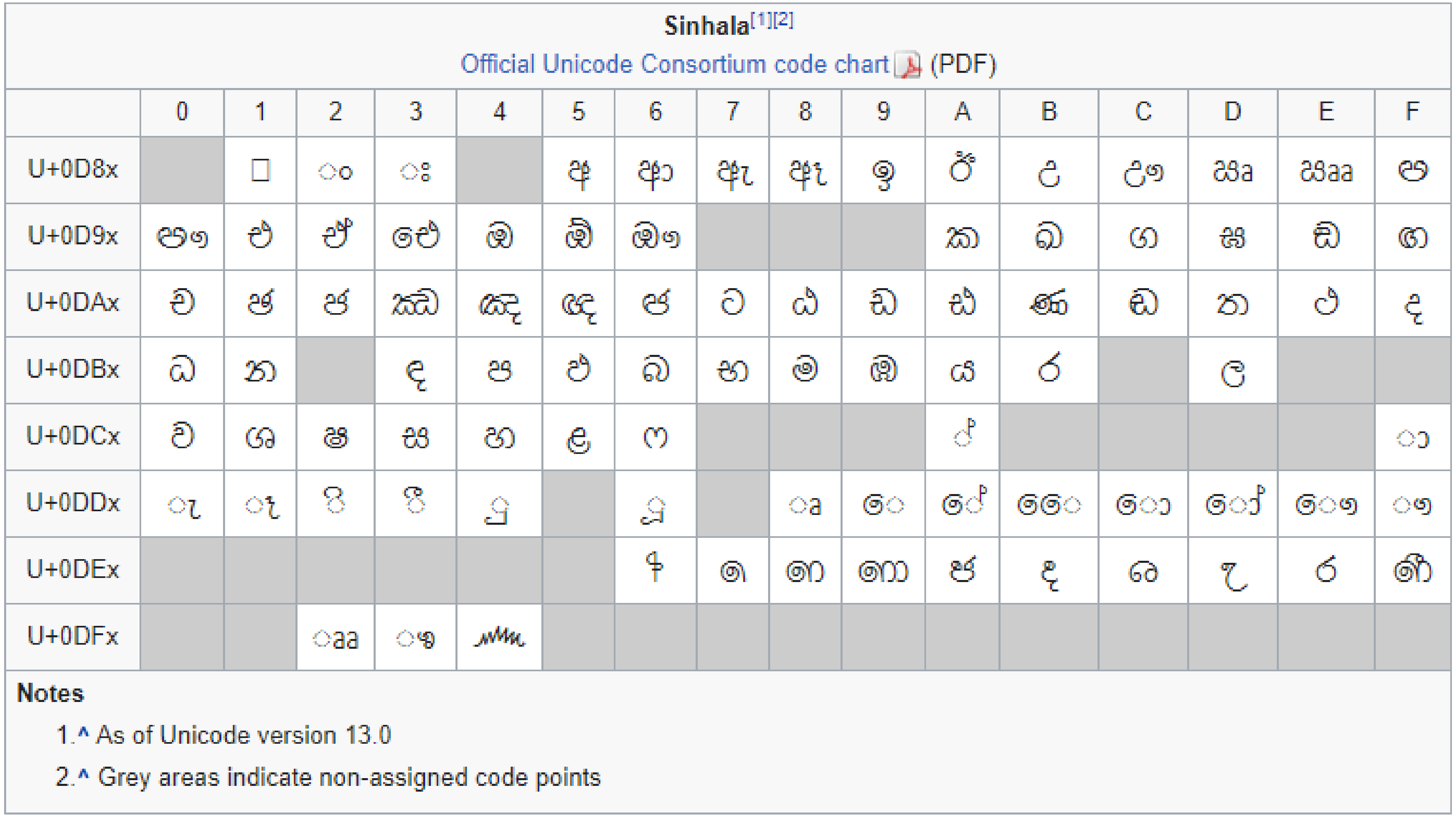
	0	1	2	3	4	5	6	7	8	9
0	nul	soh	stx	etx	eot	enq	ack	bel	bs	ht
1	nl	vt	ff	cr	so	si	dle	dc1	dc2	dc3
2	dc4	nak	syn	etb	can	em	sub	esc	fs	gs
3	rs	us	sp	!	"	#	$	%	&	'
4	(	)	*	+	,	-	.	/	0	1
5	2	3	4	5	6	7	8	9	:	;
6	<	=	>	?	@	A	B	C	D	E
7	F	G	H	I	J	K	L	M	N	O
8	P	Q	R	S	T	U	V	W	X	Y
9	Z	[	\	]	^	_	`	a	b	c
10	d	e	f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

https://www.oreilly.com/library/view/javatm-for-programmers/9780137018529/apb.html

UNICODE Character set

- Extended ASCII nor sufficient for international use
- UNICODE → uses 16-bit per character (code point)
- First 256 are same as extended ASCII character set
- Newer version uses 32-bit code unit per code point
    - supports even emojis

Sinhala Unicode Block



Sinhala[1][2]

Official Unicode Consortium code chart (PDF)

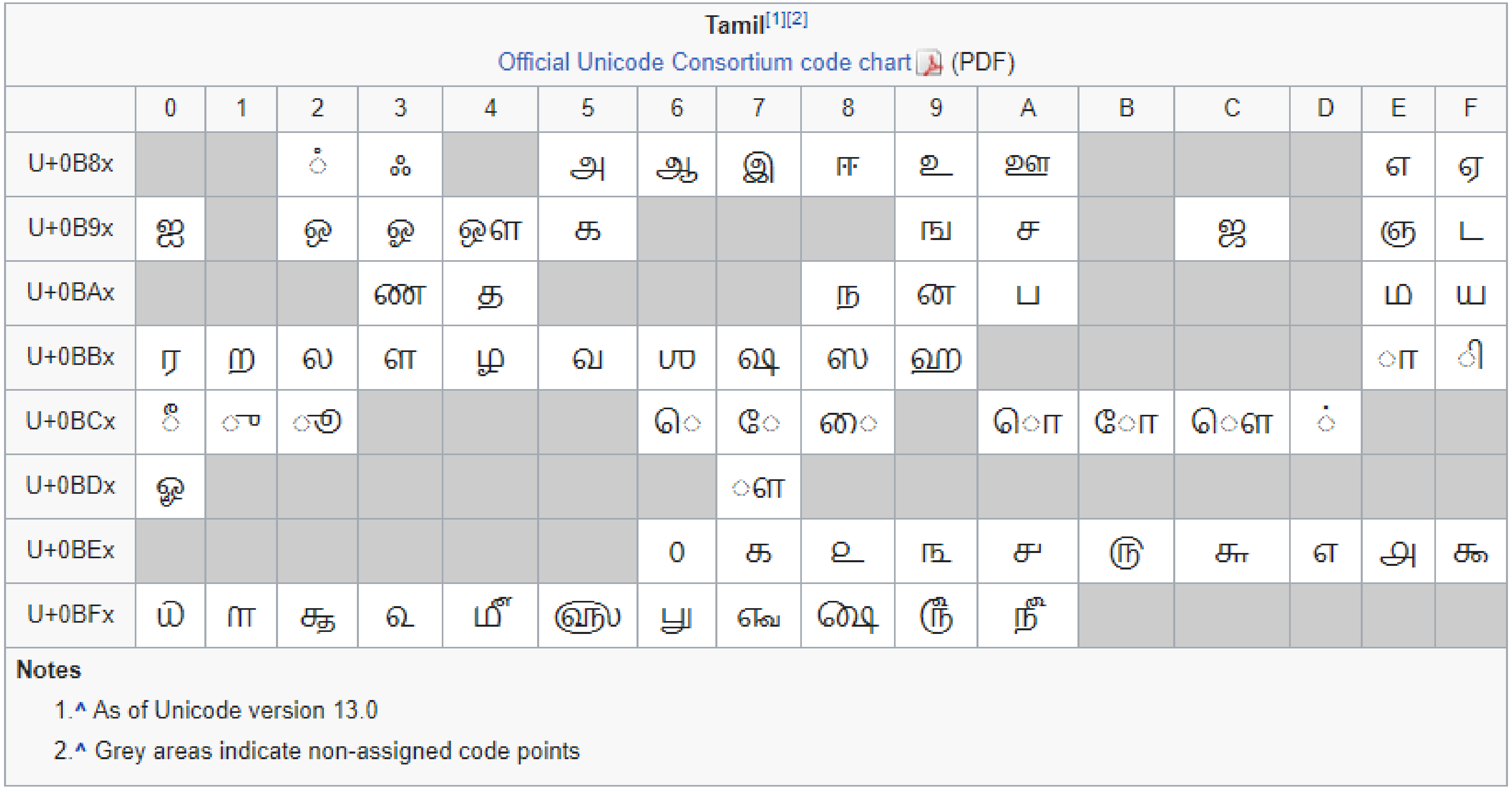
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
U+0D8x		□	◌ං	◌ඃ		අ	ආ	ඇ	ඈ	ඉ	ඊ	උ	ඌ	ඍ	ඎ	ඏ
U+0D9x	ඐ	එ	ඒ	ඓ	ඔ	ඕ	ඖ				ක	ඛ	ග	ඝ	ඞ	ඟ
U+0DAx	ච	ඡ	ජ	ඣ	ඤ	ඥ	ඦ	ට	ඨ	ඩ	ඪ	ණ	ඬ	ත	ථ	ද
U+0DBx	ධ	න		ඳ	ප	ඵ	බ	භ	ම	ඹ	ය	ර		ල		
U+0DCx	ව	ශ	ෂ	ස	හ	ළ	ෆ				◌්					◌ා
U+0DDx	◌ැ	◌ෑ	◌ි	◌ී	◌ු		◌ූ		◌ෘ	ෙ◌	◌ේ	ෛ◌	◌ො	◌ෝ	◌ෞ	◌ෟ
U+0DEx							෦	෧	෨	෩	෪	෫	෬	෭	෮	෯
U+0DFx			◌ෲ	◌ෳ	෴											

Notes

1. ^ As of Unicode version 13.0
2. ^ Grey areas indicate non-assigned code points

https://en.wikipedia.org/wiki/Sinhala_(Unicode_block)

Tamil Unicode Block



Tamil[1][2]

Official Unicode Consortium code chart (PDF)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
U+0B8x			◌ஂ	ஃ		அ	ஆ	இ	ஈ	உ	ஊ				எ	ஏ
U+0B9x	ஐ		ஒ	ஓ	ஔ	க				ங	ச		ஜ		ஞ	ட
U+0BAx				ண	த				ந	ன	ப				ம	ய
U+0BBx	ர	ற	ல	ள	ழ	வ	ஶ	ஷ	ஸ	ஹ					◌ா	◌ி
U+0BCx	◌ீ	◌ு	◌ூ				◌ெ	◌ே	◌ை		◌ொ	◌ோ	◌ௌ	◌்		
U+0BDx	ௐ							◌ௗ								
U+0BEx							௦	௧	௨	௩	௪	௫	௬	௭	௮	௯
U+0BFx	௰	௱	௲	௳	௴	௵	௶	௷	௸	௹	௺					

Notes

1. ^ As of Unicode version 13.0
2. ^ Grey areas indicate non-assigned code points

https://en.wikipedia.org/wiki/Tamil_(Unicode_block)

Images

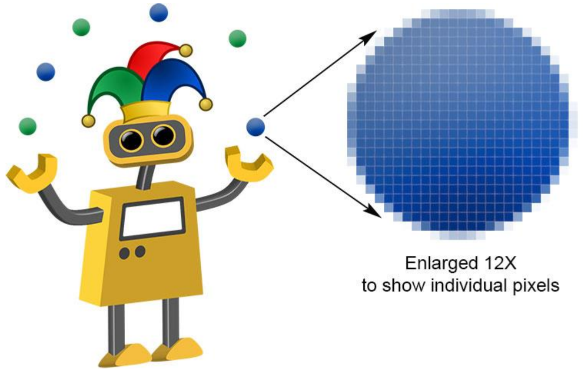
Images

Pixels (picture elements)

- Dots of color in an image (or display device)

Resolution

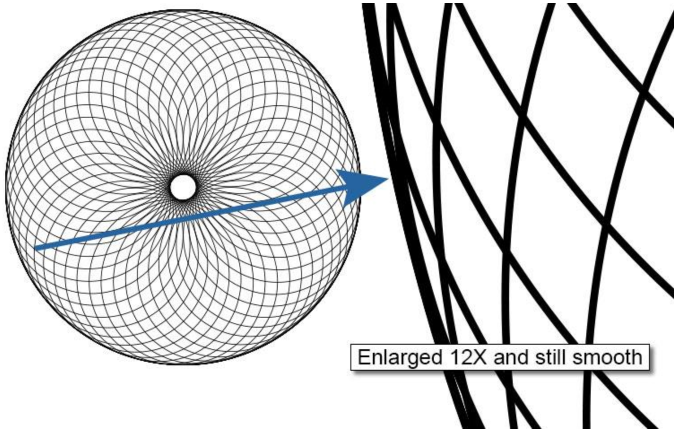
- Number of pixels in an image (or device)



Enlarged 12X
to show individual pixels

Raster graphics

- Image is represented as a collection of pixels
- Most common formats: BMP, GIF, PNG, and JPEG



Enlarged 12X and still smooth

Vector graphics

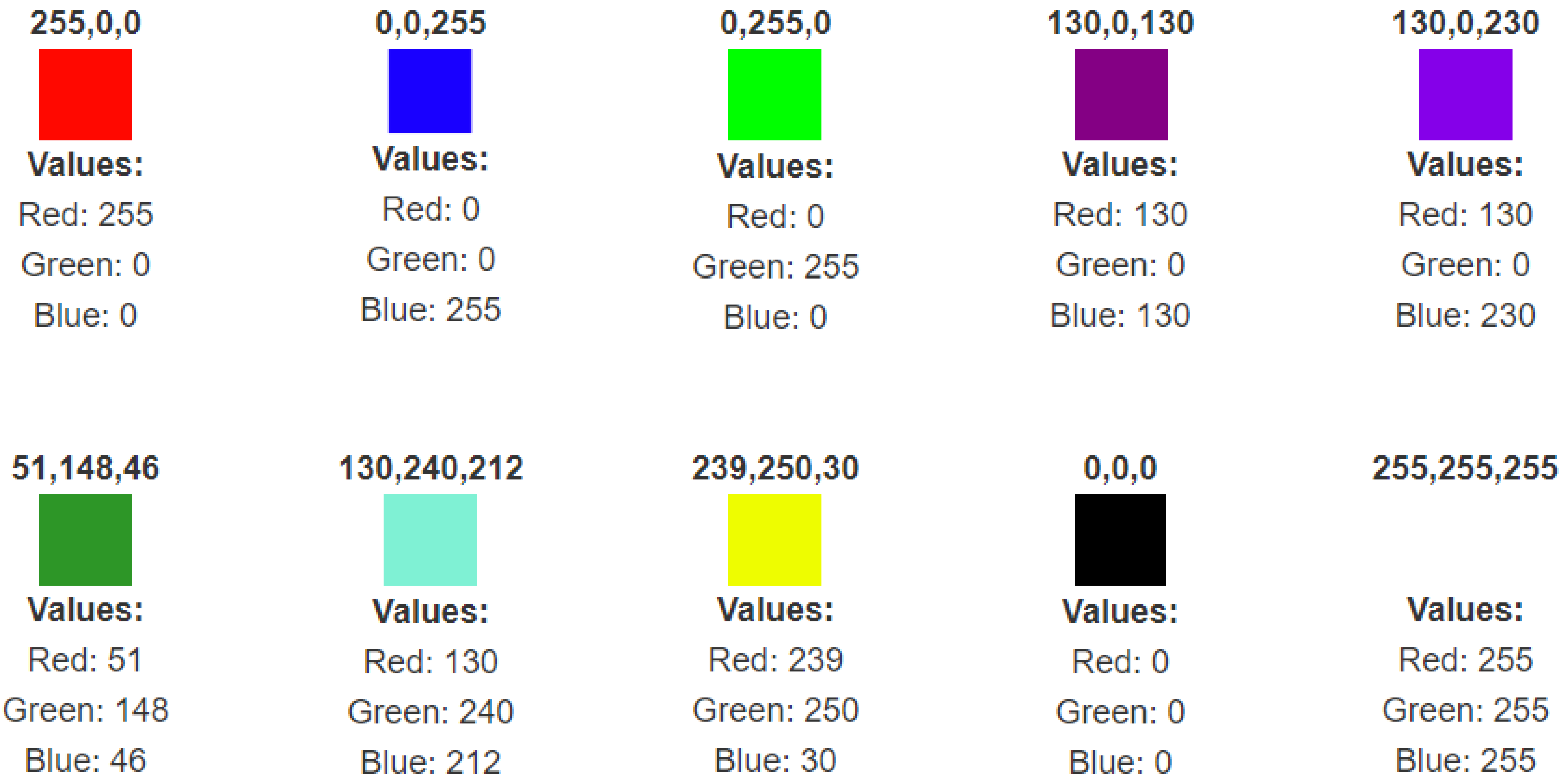
- Image is represented as a collection of geometric objects
- Most important formats: SVG

https://etc.usf.edu/techease/win/images/what-is-the-difference-between-bitmap-and-vector-images/

Bitmap - RGB

- Each pixel is coded with 3 Bytes (24-bits)
  - 1 Byte- RED
  - 1 Byte - GREEN
  - 1 Byte - BLUE

Few examples of colors within the RGB spectrum



255,0,0	0,0,255	0,255,0	130,0,130	130,0,230
Values: Red: 255 Green: 0 Blue: 0	Values: Red: 0 Green: 0 Blue: 255	Values: Red: 0 Green: 255 Blue: 0	Values: Red: 130 Green: 0 Blue: 130	Values: Red: 130 Green: 0 Blue: 230
51,148,46	130,240,212	239,250,30	0,0,0	255,255,255
Values: Red: 51 Green: 148 Blue: 46	Values: Red: 130 Green: 240 Blue: 212	Values: Red: 239 Green: 250 Blue: 30	Values: Red: 0 Green: 0 Blue: 0	Values: Red: 255 Green: 255 Blue: 255

https://www.landfx.com/docs/site-hardscape-design/site-planning/item/2672-select-color.html

Sizes of images - Bitmap

- A Bitmap image is 2048 X 1024 pixel size. What is the disk space (in MB) needed to save 50 such images? Formulate the answer.

  - No of pixel in the image = 2048 X 1024
  - each pixel coded with 3 bytes
    therefore, file size = 2048 X 1024 X 3 Bytes
  - file size in MB = $\frac{2048 \text{ X } 1024 \text{ X } 3}{1024 \text{ X } 1024}$ MB

therefore, size of 50 such images = $\frac{2048 \text{ X } 1024 \text{ X } 3}{1024 \text{ X } 1024}$ X 50 MB

Data Compression

- To reduce file size and transmission times, digital data can be compressed.

- Data compression refers to any technique that recodes the data in a file so that it contains fewer bits.

Data Compression

Compression techniques divided into two categories:

lossless and lossy

Lossless compression provides a way to compress data and reconstitute it into its original state;

uncompressed data stays exactly the same as the original data

Lossy compression throws away some of the original data during the compression process; uncompressed data is not exactly the same as the original

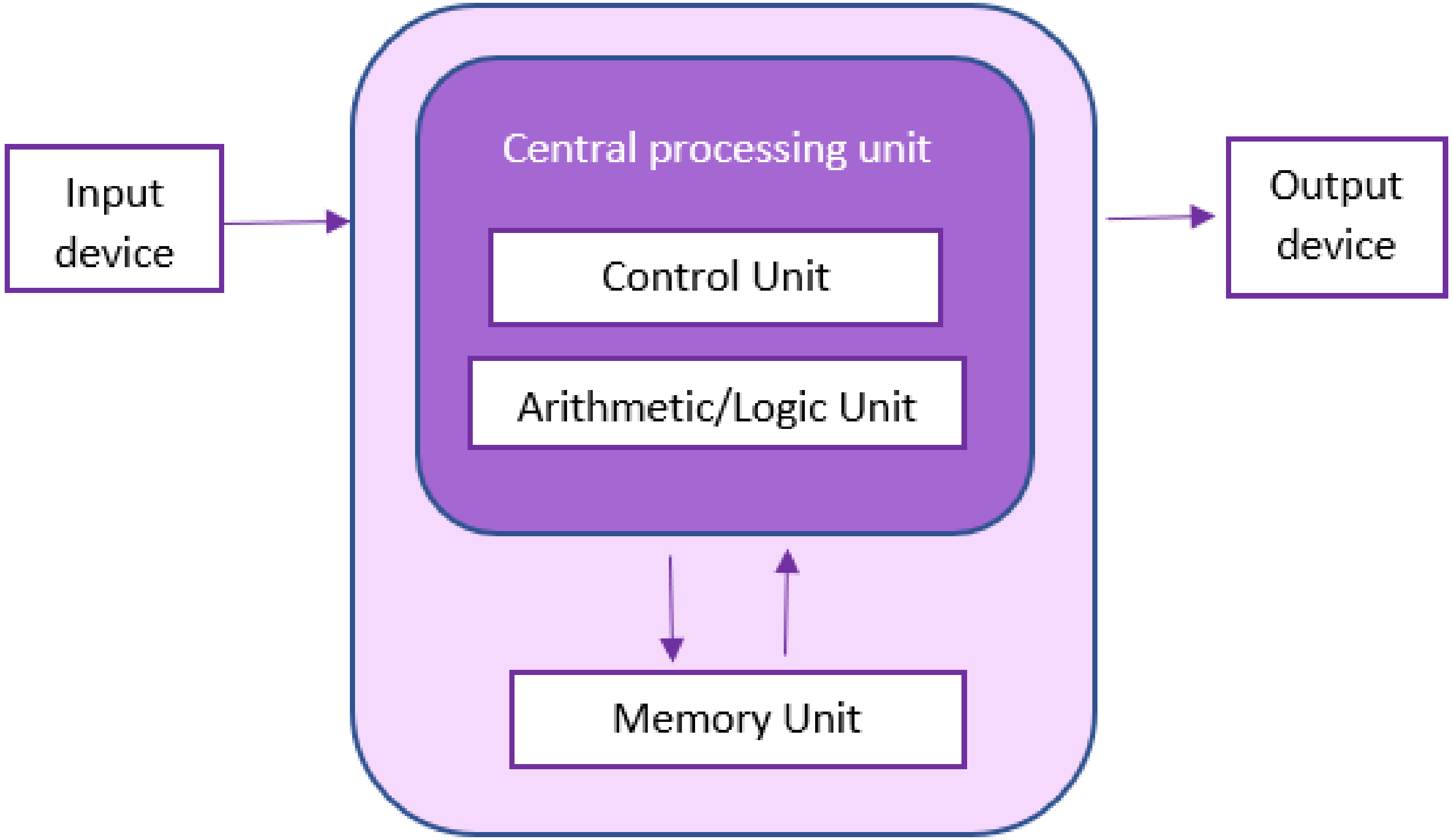
Data Encryption

- Encryption is the process of encoding messages or information in such a way that only authorized parties can read it.

- Discussions.

Computing components

The von Neumann architecture



The von Numann architecture

Arithmetic/logic unit

- Capable of performing arithmetic and logic operation

Memory Unit

- Holds data and instruction (holds the running program)

Input unit

- moves the data from outside world into the computer

Output unit

- Moves the data from inside the computer to the outside world

Control Unit

- manages all other component in the computer

System Interconnection

CPU connected to other components using buses (copper wire connections)

Address bus

- used by CPU to select which memory location or I/O or storage device to access

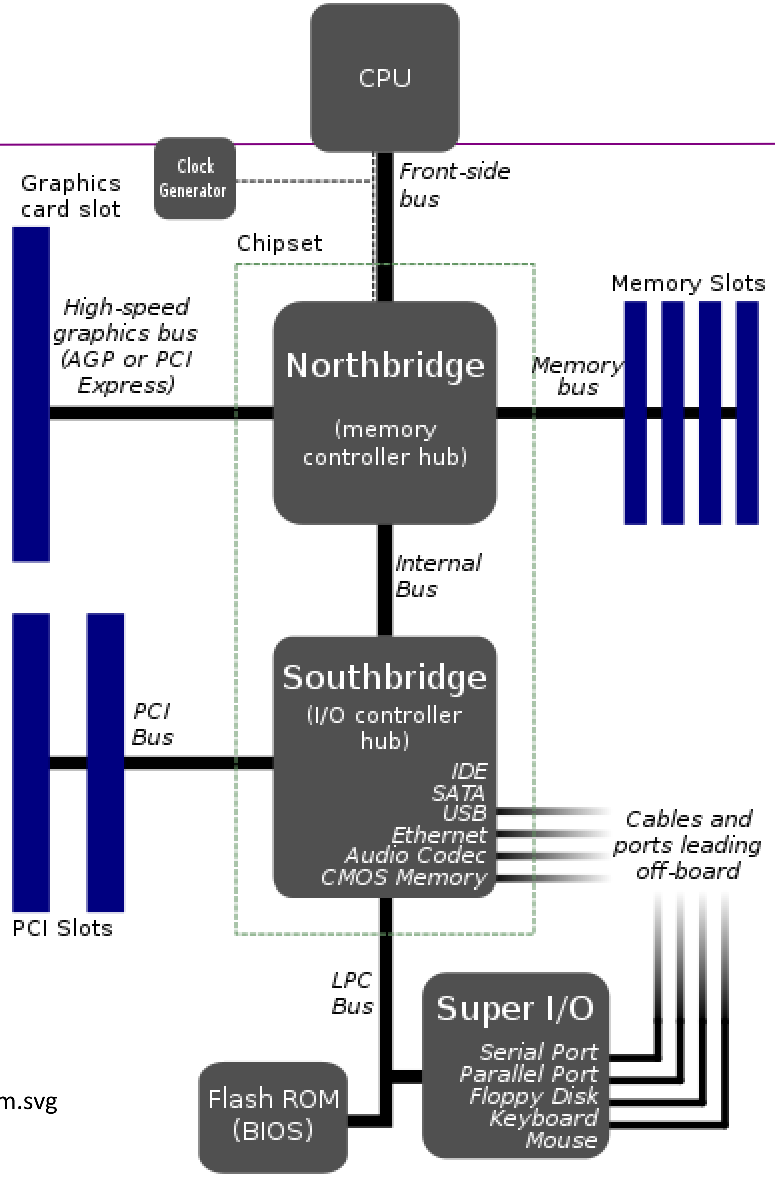
Data bus

- transmits binary data to or from CPU

Control bus

- used to control operation of memory (e.g. enable read or write control)

Modern Computers now use separate buses for more specific needs and to increase the speed of the system



- https://en.wikipedia.org/wiki/File:Motherboard_diagram.svg

RAM and ROM

RAM (random-access memory)

- memory can be accessed as well as changed
- volatile
- used as the working memory

ROM (read-only memory)

- can access the memory but not changed
- non-volatile
- used to store BIOS program

Secondary storage devices

- RAM is volatile
- Need a way to store the information
- The memory used to store the data

Eg:

magnetic tape
magnetic disk
optic disk
flash memory

REFERENCE

- Dale, N.B. and Lewis, J., 2007. Computer science illuminated. Jones & Bartlett Learning.

- https://en.wikipedia.org/wiki/File:Motherboard_diagram.svg

READING

Chapter # 3, 5

- Computer science illuminated. Jones & Bartlett Learning.