

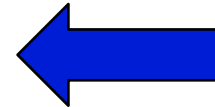
Le Digital Humanities: aspetti metodologici e pratici

Enrica Salvatori (enrica.salvatori@unipi.it)
Vittore Casarosa (casarosa@isti.cnr.it)

Pisa, 28 Marzo 2019

Refresher on Computer Fundamentals and Data Representation

- Brief History of computers
- Architecture of a computer
- Data representation within a computer
- Metadata



Early visions



Charles Babbage (1791-1871)
Professor of Mathematics at
Cambridge University (1827-1839)

Difference Engine	1823
Analytic Engine	1833

Applications

Mathematical Tables – Astronomy
and Navigation

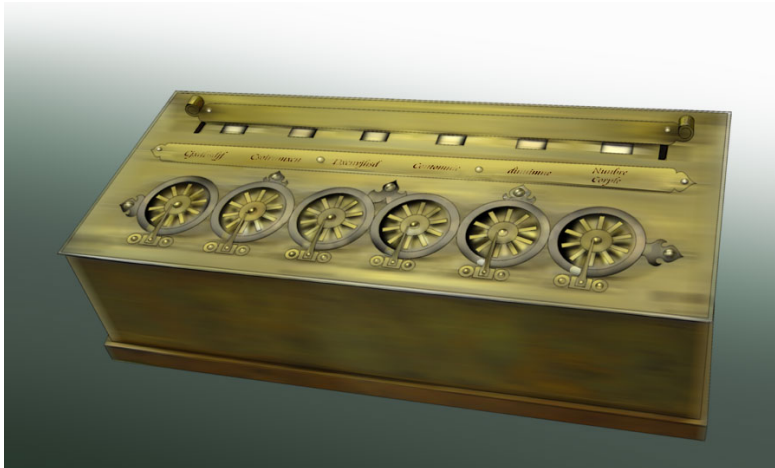
Technology

Jacquard's loom (1801) and
mechanical gears (steam
operated)

Use of punched paper tape



Pascaline (~ 1650)



Mechanical calculator
invented by Blaise Pascal
between 1640 and 1650.

It is not a computer (in our
meaning today) as it does
not have a **program**



Harvard Mark I

- Built in 1944 in IBM Endicott laboratories
 - Howard Aiken – Professor of Physics at Harvard
 - Essentially mechanical but had some electro-magnetically controlled relays and gears
 - Weighed *5 tons* and had *750,000* components
 - A synchronizing clock that beat every *0.015* seconds (66KHz)

Performance:

0.3 seconds for addition
6 seconds for multiplication
1 minute for a sine calculation

WW-2 Effort

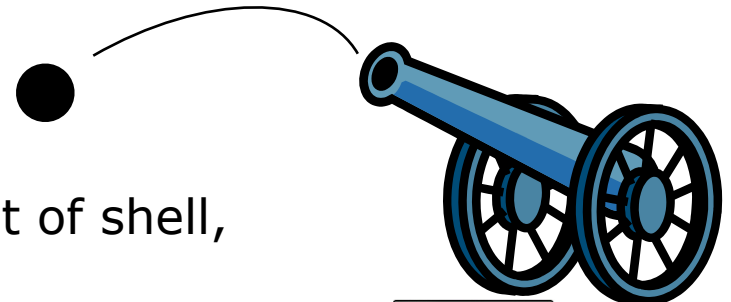
Broke down once a week!

- Inspired by Atanasoff and Berry, Eckert and Mauchly designed and built ENIAC (1943-45) at the University of Pennsylvania
- The first, completely electronic, operational, general-purpose analytical calculator!
 - 30 tons, 72 square meters, 200KW
- Performance
 - Read in 120 cards per minute
 - Addition took 200 μ s, Division 6 ms
 - 1000 times faster than Mark I
- Not very reliable!

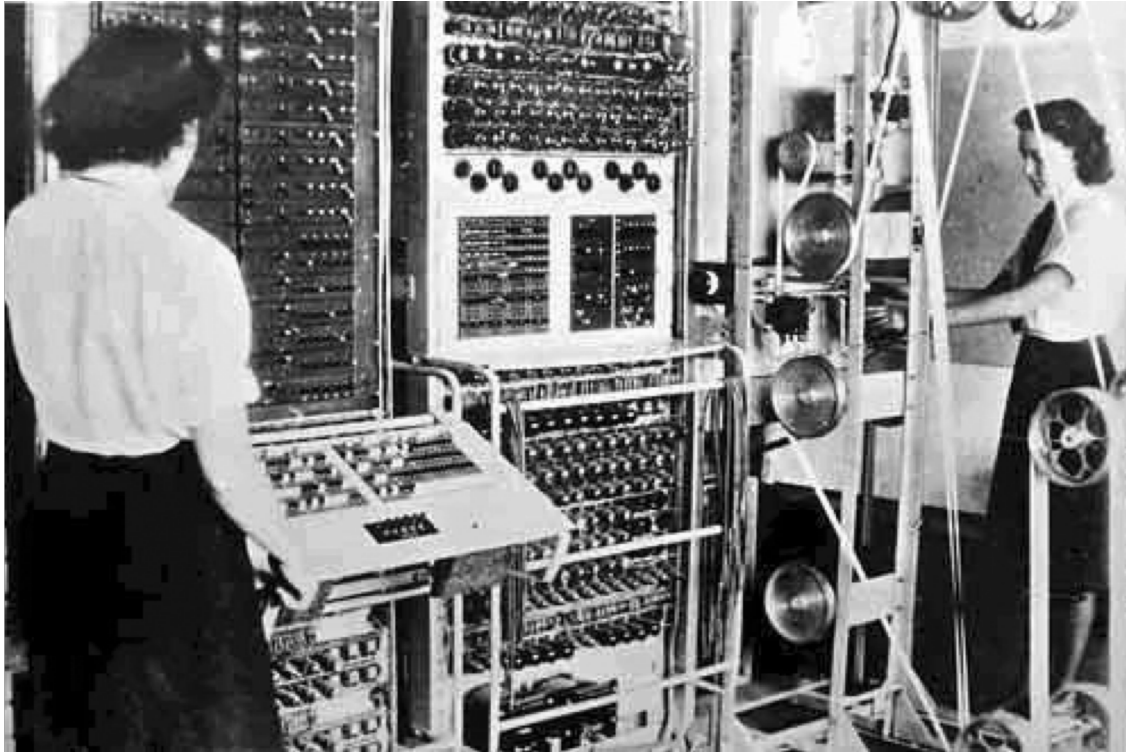
WW-2 Effort

Application: Ballistic calculations

angle = f (location, tail wind, cross wind,
air density, temperature, weight of shell,
propellant charge, ...)



Colossus



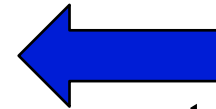
Colossus (derived from Mark 1 and Mark 2) was used in London during the second World War to decipher secret German messages (Enigma machine)

EDVAC - Electronic Discrete Variable Automatic Computer

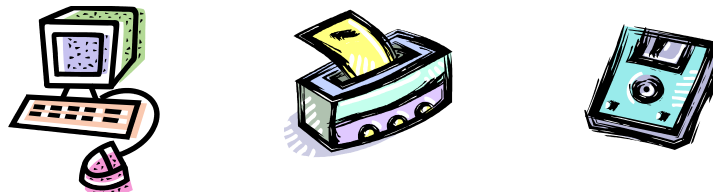
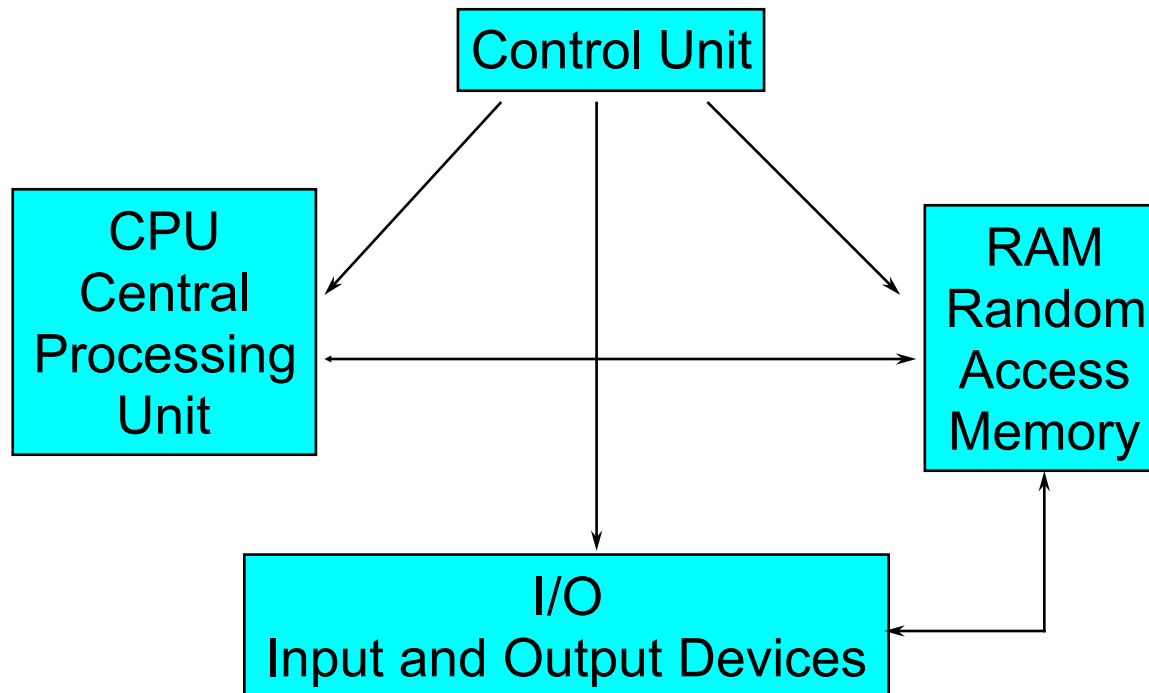
- ENIAC's programming system was external
 - Sequences of instructions were executed independently of the results of the calculation
 - Human intervention required to take instructions “out of order”
 - Eckert, Mauchly, John von Neumann and others designed EDVAC (1944) to solve this problem
 - Solution was the *stored program computer*
- ⇒ “*program can be manipulated as data*”
- *First Draft of a report on EDVAC* was **published in 1945**, but just had von Neumann's signature
 - In 1973 the court of Minneapolis attributed the honor of *inventing the computer* to John Atanasoff

Refresher on Computer Fundamentals and Data Representation

- Brief History of computers
- Architecture of a computer
- Data representation within a computer
- Metadata



Basic components of a computer

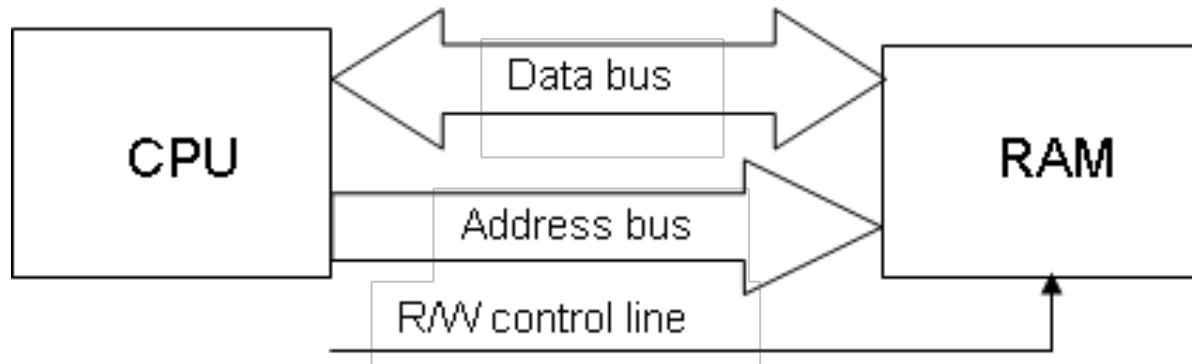


Von Neuman architecture

- The RAM contains both the program (machine instructions) and the data
- The basic model is “sequential execution”
 - each instruction is extracted from memory (in sequence) and executed
- Basic execution cycle
 - Fetch instruction (from memory) at location indicated by Location Counter
 - Increment Location Counter (to point to the next instruction)
 - Bring instruction to CPU
 - Execute instruction
 - Fetch operand from memory (if needed)
 - Execute operation
 - Store result
 - in “registers” (temporary memory in the CPU)
 - in memory (RAM)

Random Access Memory

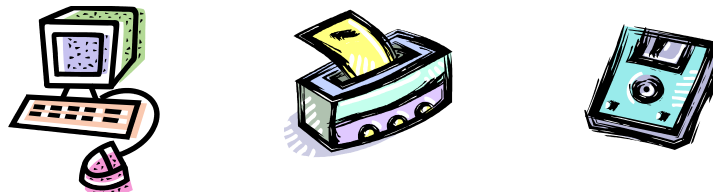
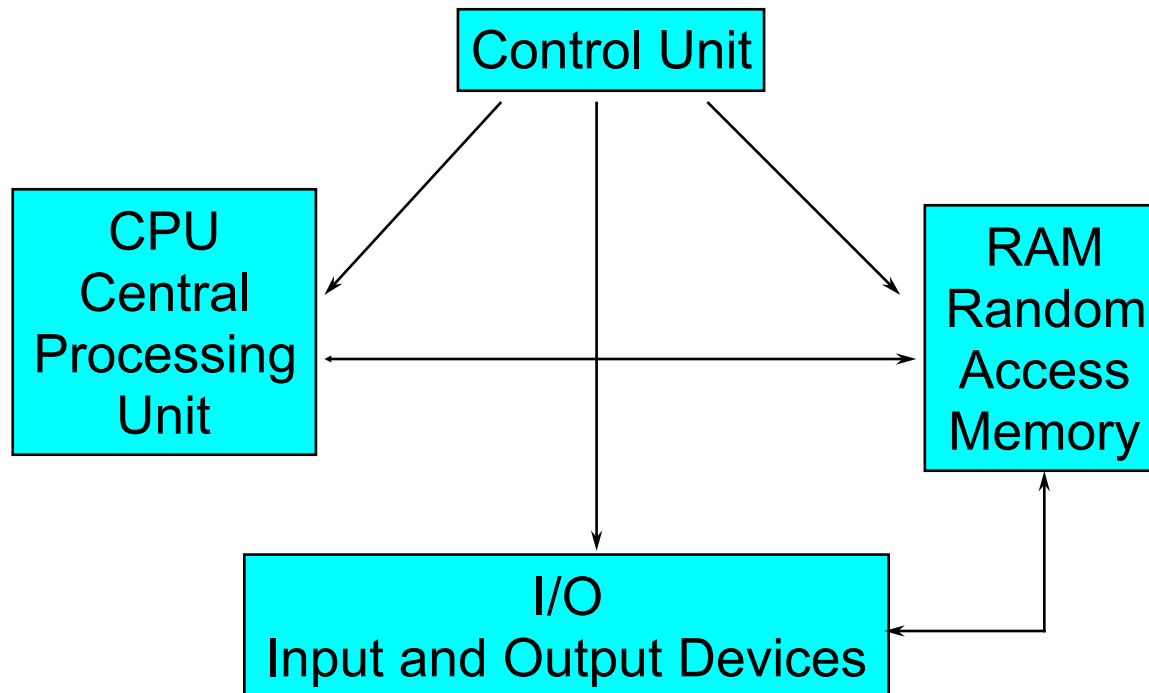
- The RAM is a linear array of “cells”, usually called “words”
- The words are numbered from 0 to N, and this number is the “address” of the word
- In order to read/write a word from/into a memory cell, the CPU has to provide its address on the “address bus”
- A “control line” tells the memory whether it is a read or write operation
- In a read operation the memory will provide on the “data bus” the content of the memory cell at the address provided on the “address bus”
- In a write operation the memory will store the data provided on the “data bus” into the memory cell at the address provided on the “address bus”



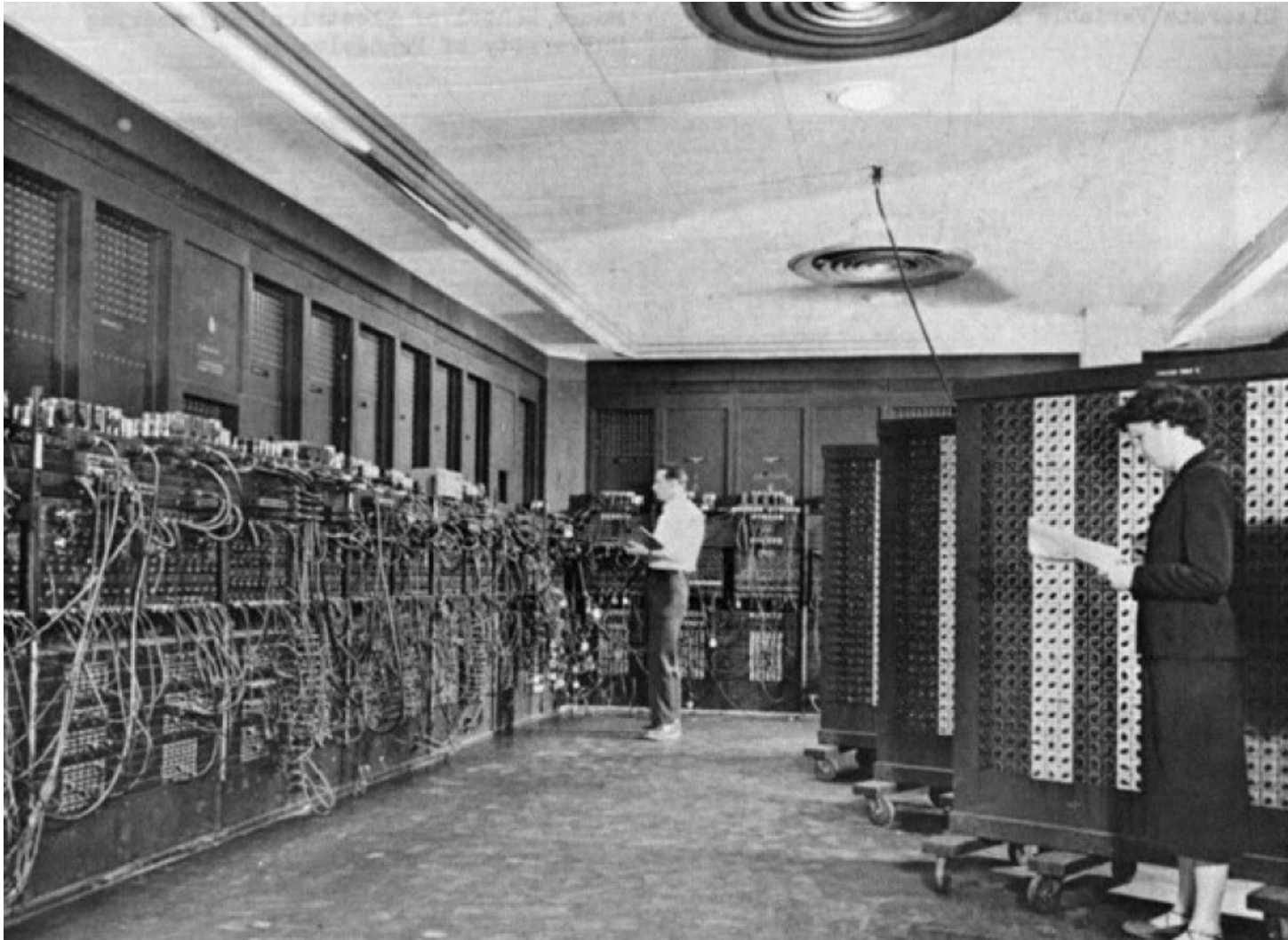
Data within a computer

- The Control Unit, the RAM, the CPU and all the physical components in a computer act on electrical signals and on devices that (basically) can be in only one of two possible states
- The two states are conventionally indicated as “zero” and “one” (0 and 1), and usually correspond to two voltage levels
- The consequence is that all the data within a computer (or in order to be processed by a computer) has to be represented with 0s and 1s, i.e. in “binary notation”

Basic components of a computer

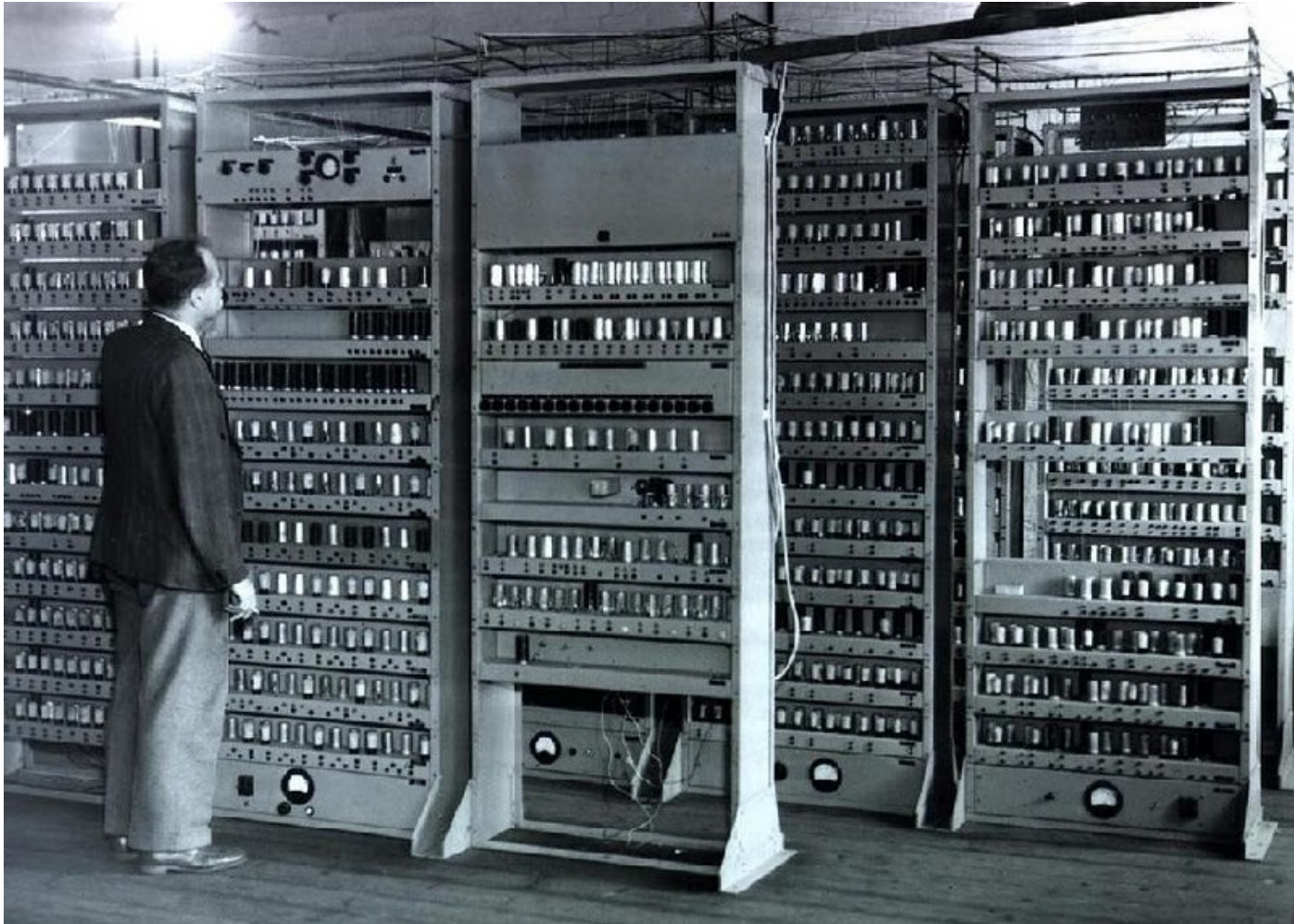


ENIAC - Electronic Numerical Integrator And Computer



EDSAC - Electronic Delay Storage Automatic Calculator

EDSAC, University of Cambridge, UK, 1949



A “mainframe” in the 60’



A “mainframe” in the 70’



Photograph: Dominic Hart/NASA Ames

Minicomputers

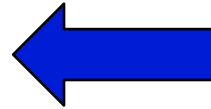


Early PCs

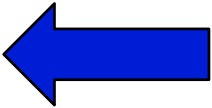


Refresher on Computer Fundamentals and Data Representation

- Brief History of computers
- Architecture of a computer
- Data representation
- Metadata



Representation of information within a computer

- Numbers 
- Text (characters and ideograms)
- Documents
- Images
- Video
- Audio

Positional notation base 10

Positional notation in base 10

Ten different symbols are needed for the digits (0,1,2,3,4,5,6,7,8,9)

The “weight” of each digit is a power of 10 (the base) and depends on its position in the number

$$10^0=1$$

$$10^1=10$$

$$10^2=100$$

$$10^3=1000$$

$$10^4=10000$$

3

4

7

$$3 \times 10^2 + 4 \times 10^1 + 7 \times 10^0 = 347$$

Roman numbers

Roman numbers are not positional

They are the sum of the values, unless a smaller value precedes a larger one; in that case the smaller value is subtracted from the larger one

I=1

XXVII = 27

V=5

XXXIV = 34

X=10

XLV = 45

L=50

MCMXCIX = 1999

C=100

MMVIII = 2008

D=500

MMIX = 2009

M=1000

MMX = 2010

Positional notation in base 8

Eight different symbols are needed for the digits (0,1,2,3,4,5,6,7)

The “weight” of each digit is a power of 8 (the base) and depends on its position in the number

$$8^0=1$$

$$8^1=8$$

$$8^2=64$$

$$8^3=512$$

$$8^4=4096$$

3

4

7

$$3 \times 8^2 + 4 \times 8^1 + 7 \times 8^0$$

$$192 + 32 + 7 = 231$$

Positional notation in base 16

Sixteen different symbols are needed for the digits (0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F)

The “weight” of each digit is a power of 16 (the base) and depends on its position in the number

$$16^0=1$$

$$16^1=16$$

$$16^2=256$$

$$16^3=4096$$

$$16^4=65536$$

3

B

F

$$3 \times 16^2 + B \times 16^1 + F \times 16^0$$

$$3 \times 256 + 11 \times 16 + 15 \times 1$$

$$768 + 176 + 15 = 959$$

Positional notation base 2

Positional notation in base 2

Two different symbols are needed for the digits (0,1)

The “weight” of each digit is a power of 2 (the base) and depends on its position in the number

$$2^0=1$$

$$2^1=2$$

$$2^2=4$$

$$2^3=8$$

$$2^4=16$$

$$2^5=32$$

$$2^6=64$$

$$2^7=128$$

$$2^8=256$$

1

0

1

1

$$1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$$

$$1 \times 8 + 0 \times 4 + 1 \times 2 + 1 \times 1$$

$$8 + 0 + 2 + 1 = 11$$

Powers of 2

$$2^0=1$$

$$2^1=2$$

$$2^2=4$$

$$2^3=8$$

$$2^4=16$$

$$2^5=32$$

$$2^6=64$$

$$2^7=128$$

$$2^8=256$$

$$2^9=512$$

$$2^{10}=1024$$

$$1K$$

$$2^{11}=2048$$

$$2K$$

$$2^{12}=4096$$

$$4K$$

$$2^{13}=8192$$

$$8K$$

$$2^{14}=16384$$

$$16K$$

$$2^{15}=32768$$

$$32K$$

$$2^{16}=65536$$

$$64K$$

.....

$$2^{20}=1.048.576$$

$$1024K$$

$$1M$$

$$2^{30}=1.073.741.824$$

$$1024M$$

$$1G$$

$$2^{32}=4.294.967.296$$

$$4096M$$

$$4G$$

Binary and hexadecimal numbers

$2^0=1$
 $2^1=2$
 $2^2=4$
 $2^3=8$
 $2^4=16$
 $2^5=32$
 $2^6=64$
 $2^7=128$
 $2^8=256$

0000=**0**
 0001=**1**
 0010=**2**
 0011=**3**
 0100=**4**
 0101=**5**
 0110=**6**
 0111=**7**

1000=**8**
 1001=**9**
 1010=**10** **A**
 1011=**11** **B**
 1100=**12** **C**
 1101=**13** **D**
 1110=**14** **E**
 1111=**15** **F**

10000=**16** **10**

decimal and hexadecimal
 decimal
 hexadecimal

01011011 si può rappresentare in esadecimale come 5D

Size of digital information

1000	k	kilo
1000 ²	M	mega
1000 ³	G	giga
1000 ⁴	T	tera
1000 ⁵	P	peta
1000 ⁶	E	exa
1000 ⁷	Z	zetta
1000 ⁸	Y	yotta

1 GB = 1000 MB

1 TB = 1000 GB

1 PB = 1000 TB

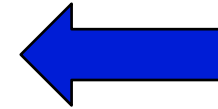
1 EB = un milione TB

1 ZB = un miliardo TB

The digital content in the world in 2018 was estimated to be about 35 zettabytes

Representation of information within a computer

- Numbers
- Text (characters and ideograms)
- Documents
- Images
- Video
- Audio



- The “natural” way to represent (alphanumeric) characters (and symbols) within a computer is to associate a character with a number, defining a “coding table”
- How many bits are needed to represent the Latin alphabet ?

The ASCII characters

! " # \$ % & ' () * + , - . /
0 1 2 3 4 5 6 7 8 9 : ; < = > ?
@ 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 [\] ^ _
` 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 95
printable
ASCII
characters,
numbered
from 32 to 126
(decimal)

33 control
characters

ASCII table (7 bits)

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	00	Null	32	20	Space	64	40	@	96	60	`
1	01	Start of heading	33	21	!	65	41	A	97	61	a
2	02	Start of text	34	22	"	66	42	B	98	62	b
3	03	End of text	35	23	#	67	43	C	99	63	c
4	04	End of transmit	36	24	\$	68	44	D	100	64	d
5	05	Enquiry	37	25	%	69	45	E	101	65	e
6	06	Acknowledge	38	26	&	70	46	F	102	66	f
7	07	Audible bell	39	27	'	71	47	G	103	67	g
8	08	Backspace	40	28	(72	48	H	104	68	h
9	09	Horizontal tab	41	29)	73	49	I	105	69	i
10	0A	Line feed	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage return	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	47	2F	/	79	4F	O	111	6F	o
16	10	Data link escape	48	30	0	80	50	P	112	70	p
17	11	Device control 1	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	50	32	2	82	52	R	114	72	r
19	13	Device control 3	51	33	3	83	53	S	115	73	s
20	14	Device control 4	52	34	4	84	54	T	116	74	t
21	15	Neg. acknowledge	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	54	36	6	86	56	V	118	76	v
23	17	End trans. block	55	37	7	87	57	W	119	77	w
24	18	Cancel	56	38	8	88	58	X	120	78	x
25	19	End of medium	57	39	9	89	59	Y	121	79	y
26	1A	Substitution	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	59	3B	;	91	5B	[123	7B	{
28	1C	File separator	60	3C	<	92	5C	\	124	7C	
29	1D	Group separator	61	3D	=	93	5D]	125	7D	}
30	1E	Record separator	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	63	3F	?	95	5F	_	127	7F	□

- ASCII 7 bits (late fifties)
 - American Standard Code for Information Interchange
 - 7 bits for 128 characters (Latin alphabet, numbers, punctuation, control characters)
- EBCDIC (early sixties)
 - Extended Binary Code Decimal Interchange Code
 - 8 bits; defined by IBM in early sixties, still used and supported on many computers
- ASCII 8 bits (ISO 8859-xx) extends original ASCII to 8 bits to include accented letters and non Latin alphabets (e.g. Greek, Russian)
- UNICODE or ISO-10646 (1993)
 - Merged efforts of the Unicode Consortium and ISO
 - UNiversal CODE still evolving
 - It incorporates all(?) the pre-existing representation standards
 - Basic rule: round trip compatibility
 - Side effect is multiple representations for the same character

ISO-8859-xx (ASCII 8-bits)

- Developed by ISO (International Organization for Standardization)
- There are 16 different tables coding characters with 8 bit
- Each table includes ASCII (7 bits) in the lower part and other characters in the upper part for a total of 191 characters and 32 control codes
- It is also known as ISO-Latin–xx (includes all the characters of the “Latin alphabet”)

ISO-8859-xx code pages

- 8859-1 Latin-1 Western European languages
- 8859-2 Latin-2 Central European languages
- 8859-3 Latin-3 South European languages
- 8859-4 Latin-4 North European languages
- 8859-5 Latin/Cyrillic Slavic languages
- 8859-6 Latin/Arabic Arabic language
- 8859-7 Latin/Greek modern Greek alphabet
- 8859-8 Latin/Hebrew modern Hebrew alphabet
- 8859-9 Latin-5 Turkish language (similar to 8859-1)
- 8859-10 Latin-6 Nordic languages (rearrangement of Latin-4)
- 8859-11 Latin/Thai Thai language
- 8859-12 Latin/Devanagari Devanagari language (abandoned in 1997)
- 8859-13 Latin-7 Baltic Rim languages
- 8859-14 Latin-8 Celtic languages
- 8859-15 Latin-9 Revision of 8859-1
- 8859-16 Latin-10 South-Eastern European languages

- ASCII (late fifties)
 - American Standard Code for Information Interchange
 - 7 bits for 128 characters (Latin alphabet, numbers, punctuation, control characters)
- EBCDIC (early sixties)
 - Extended Binary Code Decimal Interchange Code
 - 8 bits; defined by IBM in early sixties, still used and supported on many computers
- ISO 8859-1 extends ASCII to 8 bits (accented letters, non Latin characters)
- UNICODE or ISO-10646 (1993)
 - Merged efforts of the Unicode Consortium and ISO
 - UNiversal CODE still evolving
 - It incorporates all(?) the pre-existing representation standards
 - Basic rule: round trip compatibility
 - Side effect is multiple representations for the same character

- In Unicode, the word “character” refers to the notion of the abstract form of a “letter”, in a very broad sense
 - a letter of an alphabet
 - a mark on a page
 - a symbol (in a language)
- A “glyph” is a particular rendition of a character (or composite character). The same Unicode character can be rendered by many glyphs
 - Character “a” in 12-point Helvetica, or
 - Character “a” in 16-point Times
- In Unicode each “character” has a name and a numeric value (called “code point”), indicated by U+hex value.
For example, the letter “G” has:
 - Unicode name: “LATIN CAPITAL LETTER G”
 - Unicode value: U+0047 (see ASCII codes)

Unicode representation

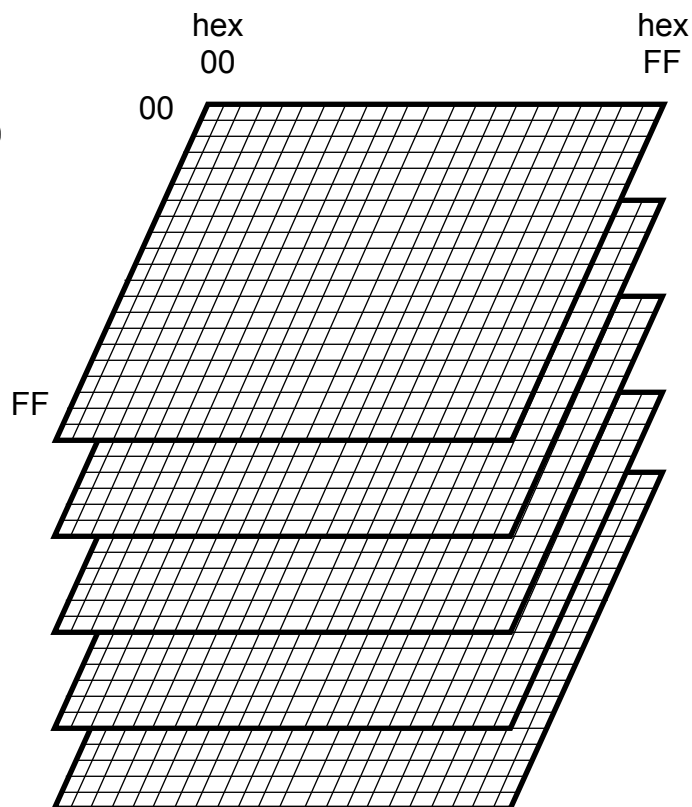
- The Unicode standard has specified (and assigned values to) about 96.000 characters
- Representing Unicode characters (code points)
 - 32 bits in ISO-10646
 - 21 bits in the Unicode Consortium
- In the 21 bit address space, there are 32 “planes” of 64K characters each (256X256)
- Only 6 planes defined as of today, of which only 4 are actually “filled”
- Plane 0, the Basic Multilingual Plane, contains most of the characters used (as of today) by most of the languages used in the Web

The planes of Unicode

256 characters (8 bits)
In each row

256 characters (8 bits)
In each column

64K characters
In each plane



Unicode planes

Plane 0	Basic Multilingual Plane	U+0000 to U+FFFF	modern languages and special characters. Includes a large number of Chinese, Japanese and Korean (CJK) characters.
Plane 1	Supplementary Multilingual Plane	U+10000 to U+1FFFF	historic scripts and musical and mathematical symbols
Plane 2	Supplementary Ideographic Plane	U+20000 to U+2FFFF	rare Chinese characters
Plane 14	Supplementary Special-purpose Plane	U+E0000 to U+EFFFF	non-recommended language tag and variation selection characters
Plane 15	Supplementary Private Use Area-A	U+F0000 to U+FFFFF	private use (no character is specified)
Plane 16	Supplementary Private Use Area-B	U+100000 to U+10FFFF	private use (no character is specified)

Unicode charts

Language characters	Kannada
Basic Latin	Khmer Symbols
Latin-1 Supplement	Khmer
Latin Extended-A	Lao
Latin Extended-B	Limbu
Latin Extended Additional	Linear B Ideograms
	Linear B Syllabary
Language specific characters	Malayalam
Alphabetic Presentation Forms	Mongolian
Arabic Presentation Forms-A	Myanmar
Arabic Presentation Forms-B	Ogham
Arabic	Old Italic
Armenian	Oriya
Bengali	Osmanya
Buhid	Runic
Cherokee	Shavian
Cypriot Syllabary	Sinhala
Cyrillic Supplement	Syriac
Cyrillic	Tagalog
Deseret	Tagbanwa
Devanagari	Tai Le
Ethiopic	Tamil
Georgian	Telugu
Gothic	Thaana
Greek and Coptic	Thai
Greek Extended	Tibetan
Gujarati	Ugaritic
Gurmukhi	Unified Canadian Aboriginal Syllabics
Hanunoo	Yi Radicals
Hebrew	Yi Syllables

Language specific characters (Chinese, Japanese, Korean)	Numbers
Bopomofo Extended	Aegean Numbers
Bopomofo	Number Forms
CJK Compatibility Forms	
CJK Compatibility Ideographs Supplement	Other symbols
CJK Compatibility Ideographs	Braille Patterns
CJK Compatibility	Byzantine Musical Symbols
CJK Radicals Supplement	Combining Diacritical Marks for Symbols
CJK Symbols and Punctuation	Control Pictures
CJK Unified Ideographs Extension A	Currency Symbols
CJK Unified Ideographs Extension B	Enclosed Alphanumerics
CJK Unified Ideographs	Letterlike Symbols
Enclosed CJK Letters and Months	Miscellaneous Technical
Hangul Compatibility Jamo	Musical Symbols
Hangul Jamo	Optical Character Recognition
Hangul Syllables	Tai Xuan Jing Symbols
Hiragana	Yijing Hexagram Symbols
Ideographic Description Characters	
Kanbun	Character modifiers and punctuation
Kangxi Radicals	Combining Diacritical Marks
Katakana Phonetic Extensions	IPA Extensions
Katakana	Phonetic Extensions
	Spacing Modifier Letters
Graphic symbols	Combining Half Marks
Arrows	General Punctuation
Block Elements	Superscripts and Subscripts
Box Drawing	
Geometric Shapes	Miscellaneous
Misc. Symbols and Arrows	Halfwidth and Fullwidth Forms
Supplemental Arrows-A	High Private Use Surrogates
Supplemental Arrows-B	High Surrogates
	Low Surrogates
Pictorial symbols	Private Use Area
Dingbats	Small Form Variants
Miscellaneous Symbols	Specials
	Supplementary Private Use Area-A
Mathematical symbols	Supplementary Private Use Area-B
Math. Alphanumeric Symbols	Tags
Math. Operators	Variation Selectors Supplement
Miscellaneous Math. Symbols-A	Variation Selectors
Miscellaneous Math. Symbols-B	
Supplemental Math. Operators	

Beginning of BMP

in this table each “column” represents 16 characters

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	C0 Controls		<u>Basic Latin</u>						C1 Controls		<u>Latin 1 Supplement</u>					
01	<u>Latin Extended-A</u>								<u>Latin Extended-B</u>							
02	<u>Latin Extended-B</u>				<u>IPA Extensions</u>						<u>Spacing Modifiers</u>					
03	<u>Combining Diacritics</u>						<u>Greek</u>									
04	<u>Cyrillic</u>															
05	<u>Cyrillic Sup.</u>		<u>Armenian</u>						<u>Hebrew</u>							
06	<u>Arabic</u>															
07	<u>Syriac</u>				<u>Arabic Sup.</u>				<u>Thaana</u>				<u>N'Ko</u>			
08	<u>(Samaritan)</u>		<u>(Mandaic)</u>		???	???	???	???	ﺀArabic Extended-A?							
09	<u>Devanagari</u>								<u>Bengali</u>							
0A	<u>Gurmukhi</u>								<u>Gujarati</u>							
0B	<u>Oriya</u>								<u>Tamil</u>							
0C	<u>Telugu</u>								<u>Kannada</u>							
0D	<u>Malayalam</u>								<u>Sinhala</u>							
0E	<u>Thai</u>								<u>Lao</u>							
0F	<u>Tibetan</u>															
10	<u>Myanmar</u>										<u>Georgian</u>					

Unicode encoding

- UTF-32 (fixed length, four bytes)
 - UTF stands for “UCS Transformation Format” (UCS stands for “Unicode Character Set”)
 - UTF-32BE and UTF-32LE have a “byte order mark” to indicate “endianness”
- UTF-16 (variable length, two bytes or four bytes)
 - All characters in the BMP represented by two bytes
 - The 21 bits of the characters outside of the BMP are divided in two parts of 11 and 10 bits; to each part is added an offset to bring it in the “surrogate zone” of the BMP (low surrogate at D800 and high surrogate at DC800)
 - in other words, they are represented as two characters in the BMP
 - UTF-16BE and UTF-16LE to indicate “endianness”
- UTF-8 (variable length, most often one byte)
 - Characters in the 7-bit ASCII represented by one byte
 - Variable length encoding (2, 3 or 4 bytes) for all other characters

Unicode example

First four characters of Welcome

Unicode

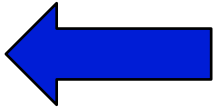
Welcome	(English)	U+0057	U+0065	U+006C	U+0063	...
Haere mai	(Māori)	U+0048	U+0061	U+0065	U+0072	...
Wilkommen	(German)	U+0057	U+0069	U+006C	U+006B	...
Bienvenue	(French)	U+0042	U+0069	U+0065	U+006E	...
Akwäba	(Fante from Ghana)	U+0041	U+006B	U+0077	U+00E4	...

	UTF-32				UTF-16				UTF-8			
Welcome	00000057	00000065	0000006C	00000063	...	0057	0065	006C	0063	...	57656C63	...
Haere mai	00000048	00000061	00000065	00000072	...	0048	0061	0065	0072	...	48616572	...
Wilkommen	00000057	00000069	0000006C	0000006B	...	0057	0069	006C	006B	...	57696C6B	...
Bienvenue	00000042	00000069	00000065	0000006E	...	0042	0069	0065	006E	...	4269656E	...
Akwäba	00000041	0000006B	00000077	000000E4	...	0041	006B	0077	00E4	...	416B77C3A4...	...

Table 4.3 Encoding the Unicode character set as UTF-8.

Unicode value	21-bit binary code	UTF-8 code			
U+00000000 – U+0000007F	0000000000000000wwwwwww	0wwwwwww			
U+00000080 – U+000007FF	0000000000wwwwxxxxxx	110wwww	10xxxxxx		
U+00000800 – U+0000FFFF	00000wwwwxxxxxxyyyyyy	1110www	10xxxxxx	10yyyyyy	
U+00010000 – U+001FFFFF	wwwxxxxxxxxxyyyyyyzzzzzz	11110ww	10xxxxxx	10yyyyyy	10zzzzzz

Representation of information within a computer

- Numbers
- Text (characters and ideograms)
- Documents 
- Images
- Video
- Audio

The editors

- Text processing applications started already in the early days of the computers (sixties)
- A “text processor” (or editor) has two main functions:
 - processing the text (delete, replace, insert, etc.)
 - specifying the format (bold, center, new line, etc.)
- The first editors were using a “mark up” language (i.e. commands intermixed with the text) to provide formatting instructions (only limited interactivity available through typewriter-like terminals)
- The “second generation” editors were using the WYSIWYG paradigm: What You See Is What You Get (much better interactivity available with display and mouse)

Representing documents

- Plain text
 - No information about structure
 - Different representation for line breaks
 - Windows represent a new line with the sequence “carriage return” followed by “line feed”
 - Unix and Apple/Mac represent a new line with “line feed” only
- Page description languages
 - PostScript
 - PDF – Portable Document Format
- Word processors
 - RTF – Rich Text Format
 - Microsoft Word
 - LaTeX

PostScript

- First commercially available page description language (Adobe 1985)
- It is a real programming language (variables, procedures, etc.) and a PostScript document is actually a “PostScript program”
- A page description comprises a number of graphical drawing instructions, including those that draw letters in a specific font in a specific size
 - Type-1 (Adobe) fonts versus TrueType (Apple)
- The document can be printed (or displayed) by having a “PostScript interpreter” executing the program
- The “abstract” PostScript description is converted to a matrix of dots (“rasterization” or “rendering”)
- PostScript initially designed for printing
 - Photo typesetters resolution up to 12000 dpi (dots per inch)
- PostScripts documents in a Digital Library
 - Extraction of text not always immediate
 - Digital Library must have a PostScript interpreter

PDF

Portable Document Format

- Successor to PostScript, to include good support for displays
- No longer a real programming language
- It defines an overall structure for a pdf document
 - Header, objects, cross-references, trailer
- Support for interactive display
 - Hierarchically structured content
 - Random access to pages
 - Navigation within a document
 - Support of hyperlinks
 - Support of “searchable images”
 - Limited editing capabilities

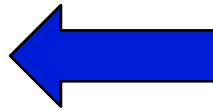
- Dates back to 1987
- Designed primarily to exchange documents among different word processors
- Description must allow a word processor to change “everything” (fonts, typesetting, tables, graphics, etc.)
- It defines an overall structure for a rtf document
 - Header, body

```
{\rtf1\ansi\deff0{\fonttbl{\f0\froman Times;}{\f1\fswiss Helvetica;}}
{\info{\title Welcome example}{\creatim\yr2001\mo8\dy10}{\nofpages1}
}\pard\plain\f1\fs28\uc0
Welcome
Haere mai
Wilkommen
Bienvenue
Akw\u228ba
\par}
```

- Widely used in the scientific and mathematical communities
- Based on TeX, defined in the late seventies by Don Knuth, to overcome the limitations of the typesetters available at the time
- LaTeX documents are expressed in plain text, to expose all the details of the internal representation
 - Any text editor on any platform can be used to compose LaTeX document
 - Converted to a page description language (typically PostScript or PDF) to get the formatted document
- Simple document structure
 - Preamble to set the defaults and the global features
 - Structured (sections and subsections) document content
- Highly customizable with “external packages”
- Text extraction not so immediate
 - A single document may occupy several files
 - Possibility of “too much” customization

Representation of information within a computer

- Numbers
- Text (characters and ideograms)
- Documents
- Images
- Video
- Audio



Welcome



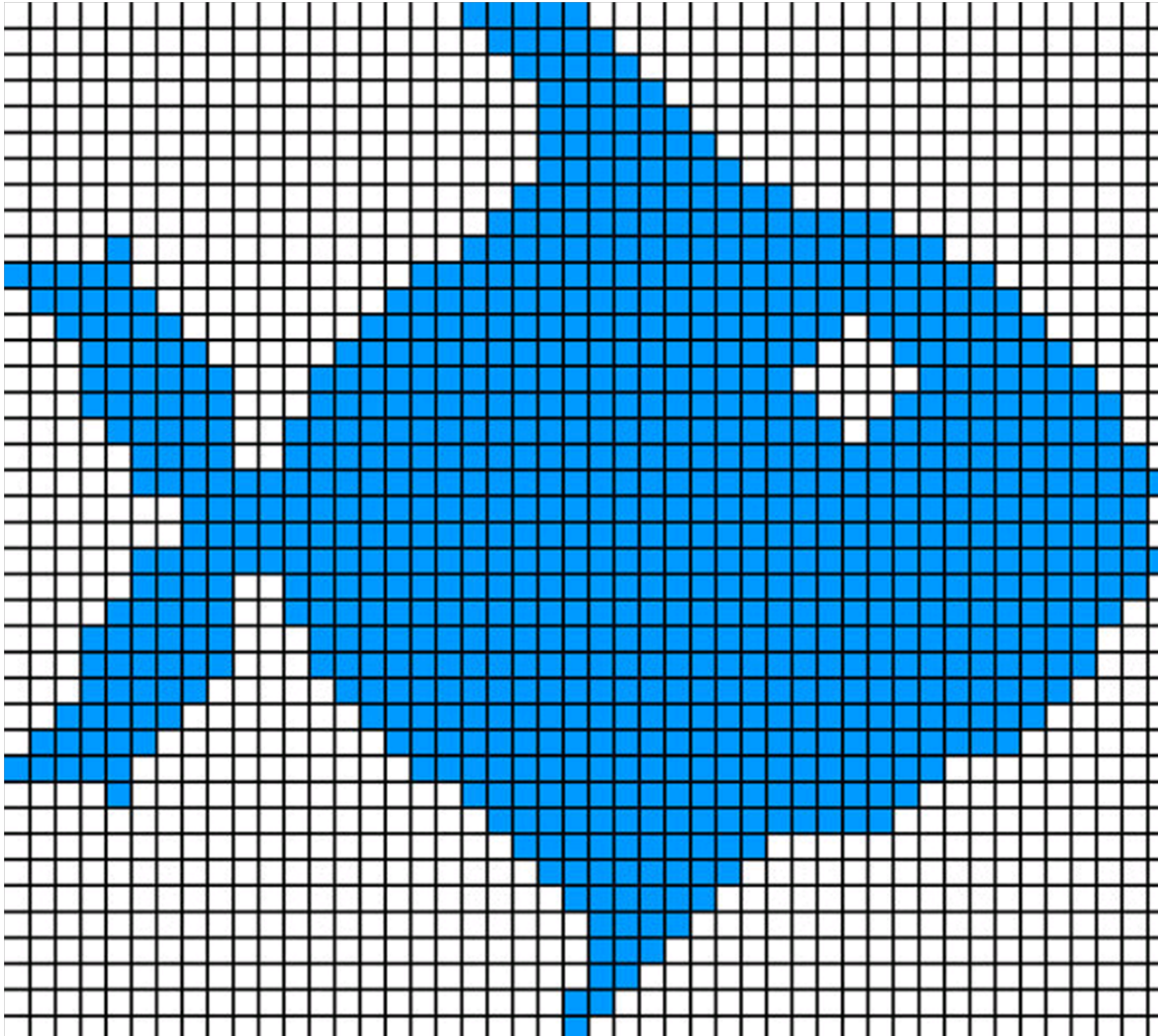
Welcome to image
representation and
compression



Representation of images

- Vector formats (geometric description)
 - Postscript
 - PDF
 - SVG (Scalable Vector Graphics)
 - SWF (ShockWave Flash)
 - from FutureWave Software to Macromedia to Adobe
 - vector-based images, plus audio, video and interactivity
 - can be played by Adobe Flash Player (browser plug-in or stand-alone)
- Raster formats (array of “picture elements”, called “pixels”)

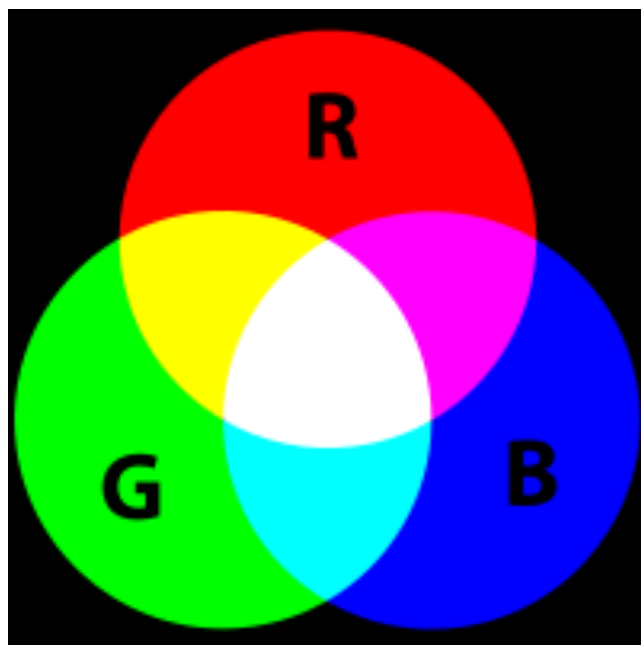
Picture elements (pixels)



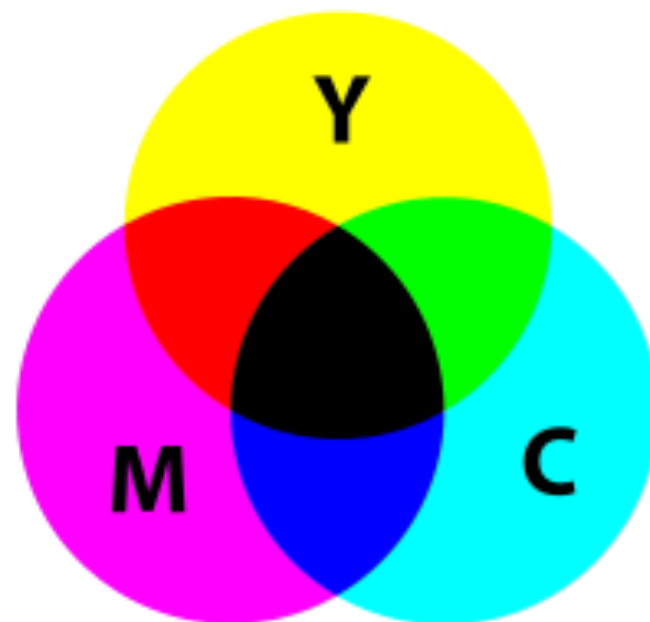
A pixel must be small enough so that its color can be considered uniform for the whole pixel. Inside the computer, a pixel is represented with a number representing its color.

Raster format

- In raster format an image (picture) is represented by a matrix of “pixels”
- Colors are represented by three numbers, one for each “color component”
- The quality of a picture is determined by:
 - The number of rows and columns in the matrix
 - Very often it is expressed as “dots per inch” (dpi)
 - 200-4800 dpi (most common ranges)
 - The number of bits representing one pixel (called depth)
 - 1 bit for black and white
 - 8-16 bits for gray scale (most common ranges)
 - 12-48 bits for color images (most common ranges)
- Big file sizes for (uncompressed color) pictures
 - For example, one color page scanned at 600 dpi is about 100 MB



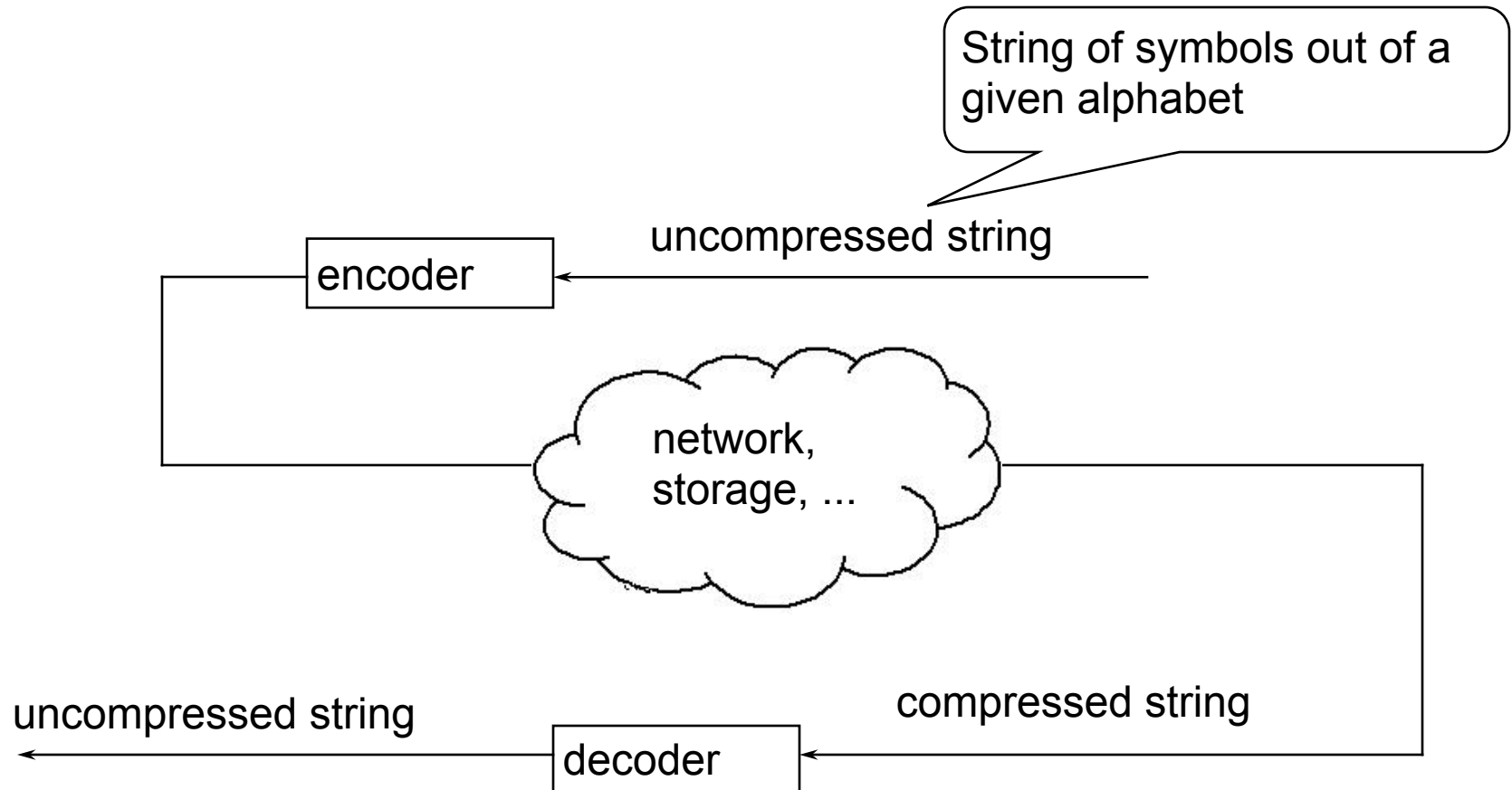
Additive color mixing



Subtractive color mixing

- Big file sizes for (uncompressed color) pictures.
Compression is (usually) needed
- Lossless compression
 - G3, G4, JBIG
 - GIF, PNG
- Lossy compression
 - JPEG
- Image containers
 - TIFF
- BMP, RAW (sensor output), DNG (Digital Negative), etc.

Compression process



Lossless data compression

- The idea of text compression, or more generally data compression, is that when the data is not needed for processing (e.g. when in transit over a network or when stored on secondary storage), then it can be represented in a more compact form (with less bits), provided that it can be brought back to the original format when needed, i.e. we want to make a “lossless compression”.
- Given a string of symbols of a given alphabet (e.g. a string of characters out of the 26 letters of the English alphabet, or a string of numbers out of the 10 digits), which is represented in the computer by N bits, the compression process takes this string and represents it in a different way so that after compression the string takes n bits, with $n < N$
- Compression is not usually noticed (which means that it is well done) but it is used in a number of applications, such as transmission of fax, downloading of web pages, transmission of data over a network, storage of data onto secondary storage, zip files, tar files, etc.

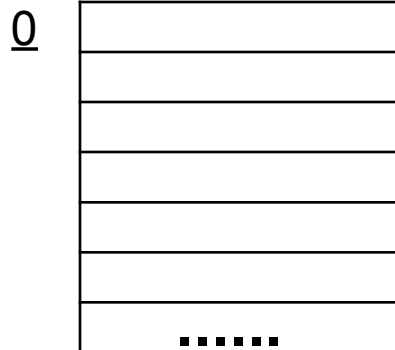
- Lossless compression
 - G3, G4, JBIG
 - GIF, PNG
- Lossy compression
 - JPEG
- Image containers
 - TIFF
- BMP, RAW (sensor output), DNG (Digital Negative), etc.

- CCITT standard (since late seventies) for fax
 - Comité Consultatif International de Télégraphie et de Téléphonie, part of ITU – International Telecommunications Union
- Specifies resolution
 - 200 x 100 dpi (standard) or 200 x 200 dpi (high resolution)
- Basically bi-level documents (black and white), even if G4 includes also provisions for optional greyscale and color images
- A one-page A4 document contains 1728x1188 pixels (bits), which is about 2 MB of data (too much to be sent over telephone lines, especially at that time)
- G3 specifies two coding (compression) methods.
 - One-dimensional (each line treated separately)
 - Two-dimensional (called READ, exploits coherence between successive scan lines)
- G4 and JBIG are more recent versions of the standard, which allow a much better compression

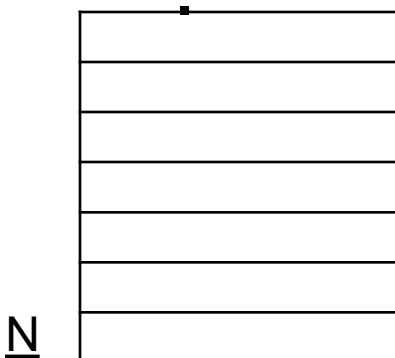
- Lossless compression
 - G3, G4, JBIG
 - GIF, PNG
- Lossy compression
 - JPEG
- Image containers
 - TIFF
- BMP, RAW (sensor output), DNG (Digital Negative), etc.

Pixel representation in GIF

image - 8 bits/pixel
sequence of rows



pointer to
color table



color table
24-36-48 bits



.....

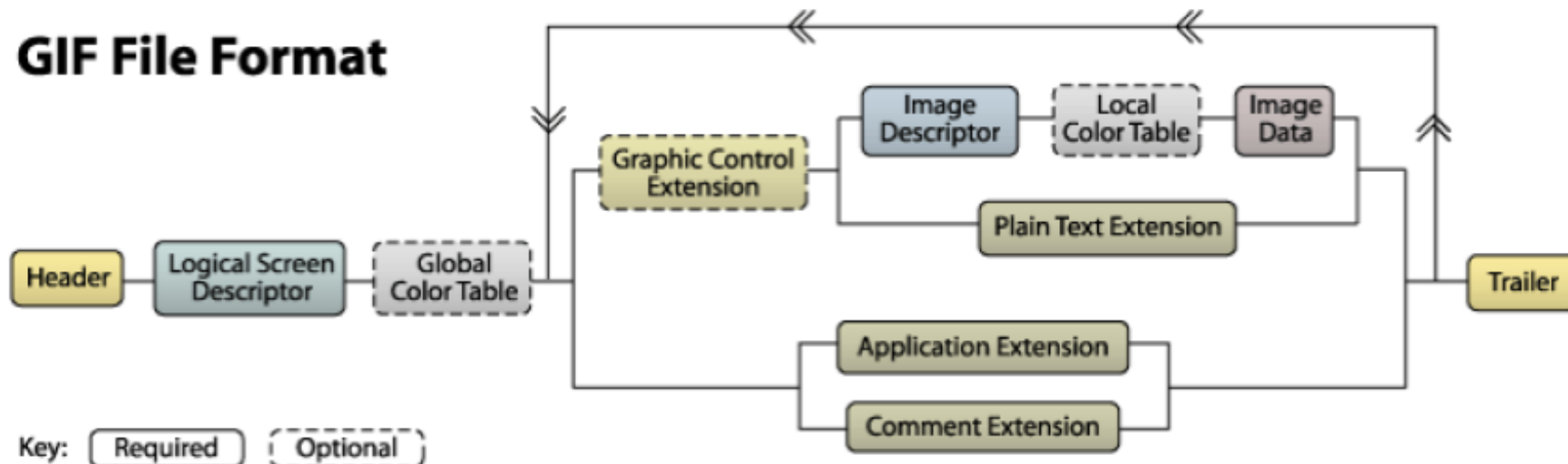
.....

.....

255



GIF File Format



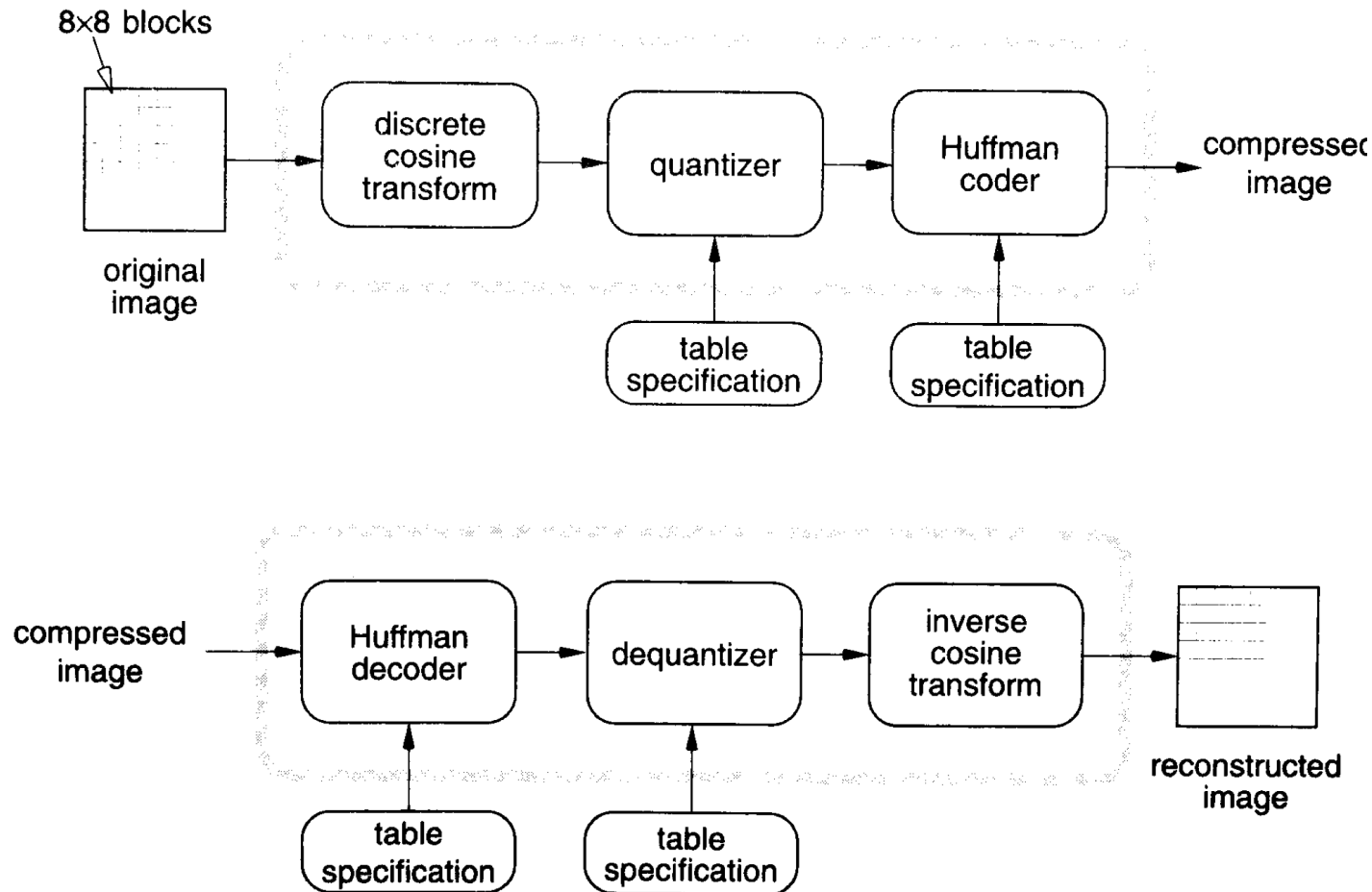
GIF and PNG

- GIF – Graphics Interchange Format, is probably the most used “lossless” compression format for images (late eighties)
- Each file may contain several images (it supports animation)
- In an image, each pixel is represented by 8 bits (or less), and the value is an index in a color table, which can be included in the file (if not included, a standard color table is used)
- The color table has 256 entries, therefore a GIF image can have a “palette” of at most 256 colors (which is much less than the colors actually in the picture)
- The pixel index values are compressed using the LZW method
- The LZW coded information is divided in blocks, preceded by a header with a byte count, so it is possible to skip over images without decompressing them
- PNG (Portable Network Graphics) is essentially the same, and was defined some years later to avoid the use of the “proprietary” LZW compression algorithm
 - PNG uses “public domain” *gzip* or *deflate* methods
 - It incorporates also several improvements over GIF

- Lossless compression
 - G3, G4, JBIG
 - GIF, PNG
- Lossy compression
 - JPEG
- Image containers
 - TIFF
- BMP, RAW (sensor output), DNG (Digital Negative), etc.

- For grayscale and color images, lossless compression still results in “too many bits”
- Lossy compression methods take advantage from the fact that the human eye is less sensitive to small greyscale or color variation in an image
- JPEG - Joint Photographic Experts Group and Joint Binary Image Group, part of CCITT and ISO
- JPEG can compress down to about one bit per pixel (starting with 8-48 bits per pixel) still having excellent image quality
 - Not very good for fax-like images
 - Not very good for sharp edges and sharp changes in color
- The encoding and decoding process is done on an 8x8 block of pixels (separately for each color component)

JPEG encoding and decoding



- Arithmetic coding instead of Huffman coding (10% improvement in compression)
- JPEG-2000 - Use of wavelets instead of DCT (20% improvement in compression, better quality for images with sharp edges)
- JPEG-LS – state of the art lossless compression
 - For each pixel, what is coded is the difference between the actual pixel value and a prediction of pixel value based on the pixel context
- Compression rates
 - 0.25–0.5 bit/pixel: moderate to good quality, sufficient for some applications
 - 0.5–0.75 bit/pixel: good to very good quality, sufficient for many applications
 - 0.75–1.5 bit/pixel: excellent quality, sufficient for most applications
 - 1.5–2 bits/pixel: usually indistinguishable from the original, sufficient for the most demanding applications

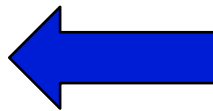
- Lossless compression
 - G3, G4, JBIG
 - GIF, PNG
- Lossy compression
 - JPEG
- Image containers
 - TIFF
- BMP, RAW (sensor output), DNG (Digital Negative), etc.

- Tagged Image File Format – file format that includes extensive facilities for **descriptive metadata**
 - note that TIFF tags are not the same thing as XML tags
- Owned by Adobe, but public domain (no licensing)
- Large number of options
 - Problems of backward compatibility
 - Problems of interoperability(Thousands of Incompatible File Formats 😊)
- Can include (and describe) four types of images
 - bilevel (black and white), greyscale, palette-color, full-color
- Support of different color spaces
- Support of different compression methods
- Much used in digital libraries and archiving

- Lossless compression
 - G3, G4, JBIG
 - GIF, PNG
- Lossy compression
 - JPEG
- Image containers
 - TIFF
- BMP, RAW (sensor output), DNG (Digital Negative), etc.

Representation of information within a computer

- Numbers
- Text (characters and ideograms)
- Documents
- Images
- Video
- Audio

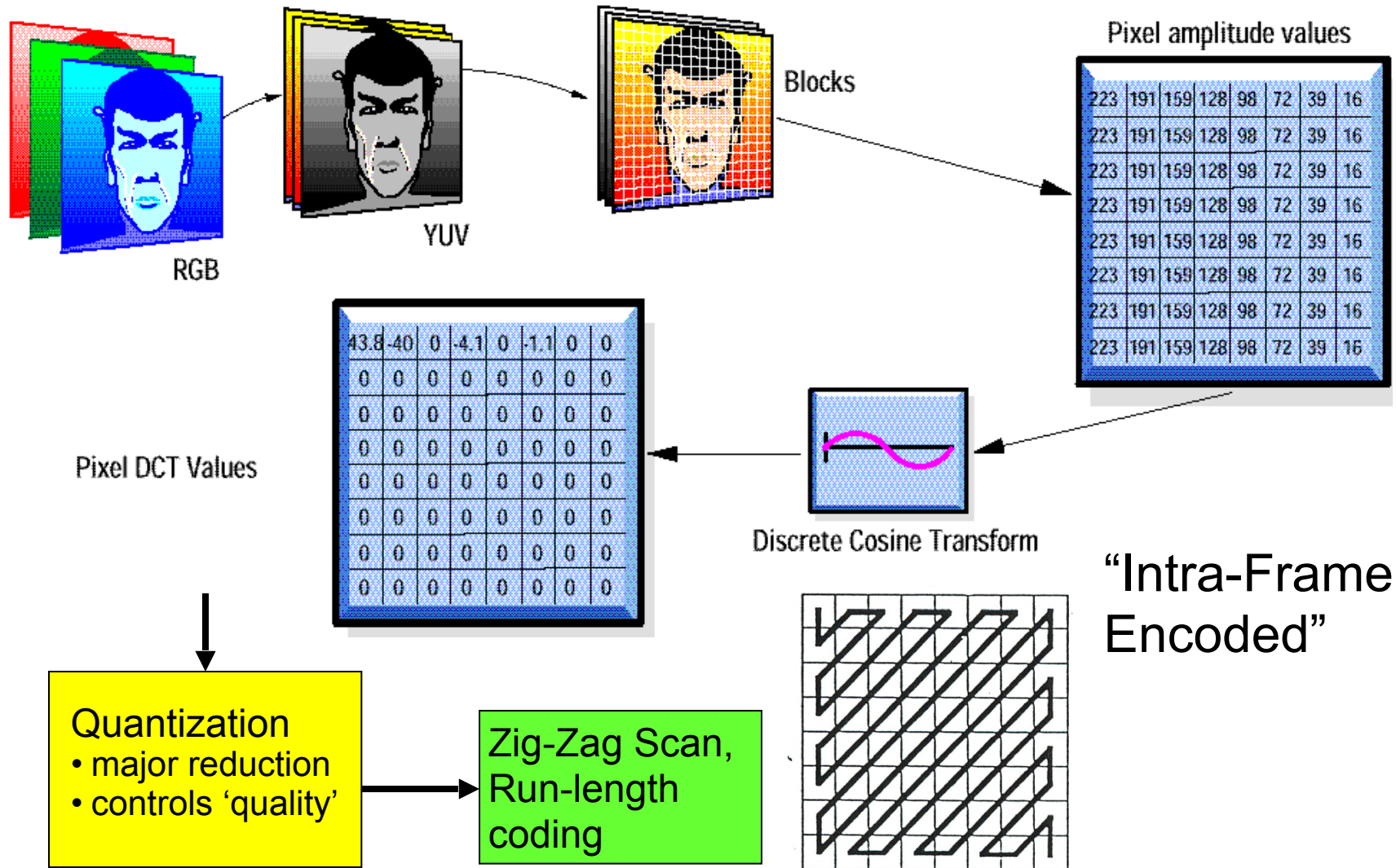


Representing video

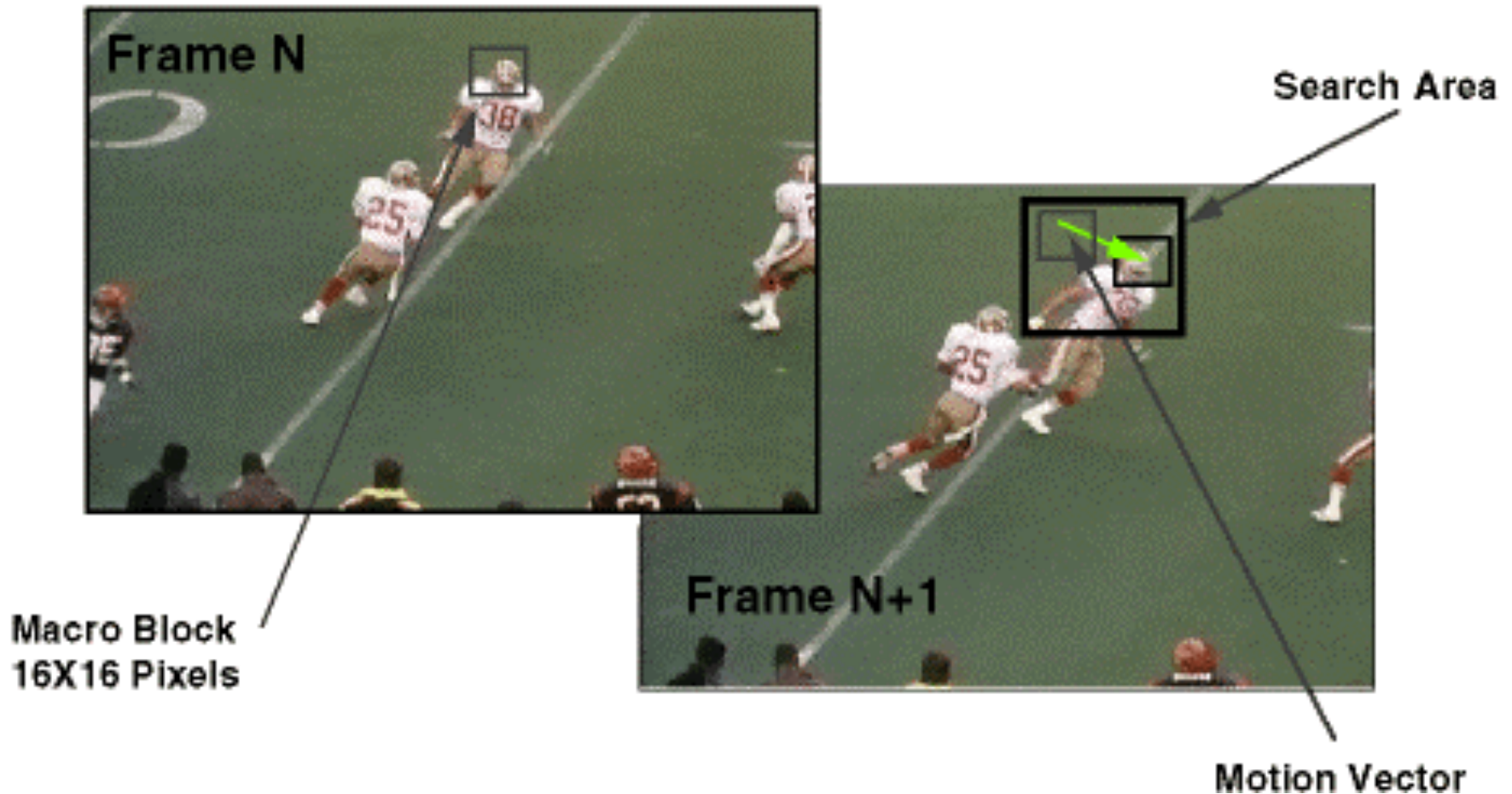
- Sequence of *frames* (still images) displayed with a given frequency
 - NTSC 30 f/s, PAL 25 f/s, HDTV 60 f/s
- Resolution of each frame depend on quality and video standard
 - 720x480 NTSC, 768x576 PAL, 1920x1080 HDTV, 3840×2160 UltraHD, 4096×2160 4K
- Uncompressed video requires “lots of bits”
 - e.g. $1920 \times 1080 \times 24 \times 30 = \sim 1,5 \text{ GB/sec}$
- It is possible to obtain very high compression rates
 - Spatial redundancy (within each frame, JPEG-like)
 - Temporal redundancy (across frames)

- MPEG - Motion Picture Experts Group established in 1988 as a committee of ISO to develop an open standard for digital TV format (CD-ROM)
- Business motivations
 - Two types of application for videos:
 - Asymmetric (encoded once, decoded many times)
 - Broadcasting, CD's
 - Video games, Video on Demand
 - Symmetric (encoded once, decoded once)
 - Video phone, video mail ...
- Design point for MPEG-1
 - Video at about 1.5 Mbits/sec
 - Audio at about 64-192 kbits/channel

Spatial Redundancy Reduction (DCT)

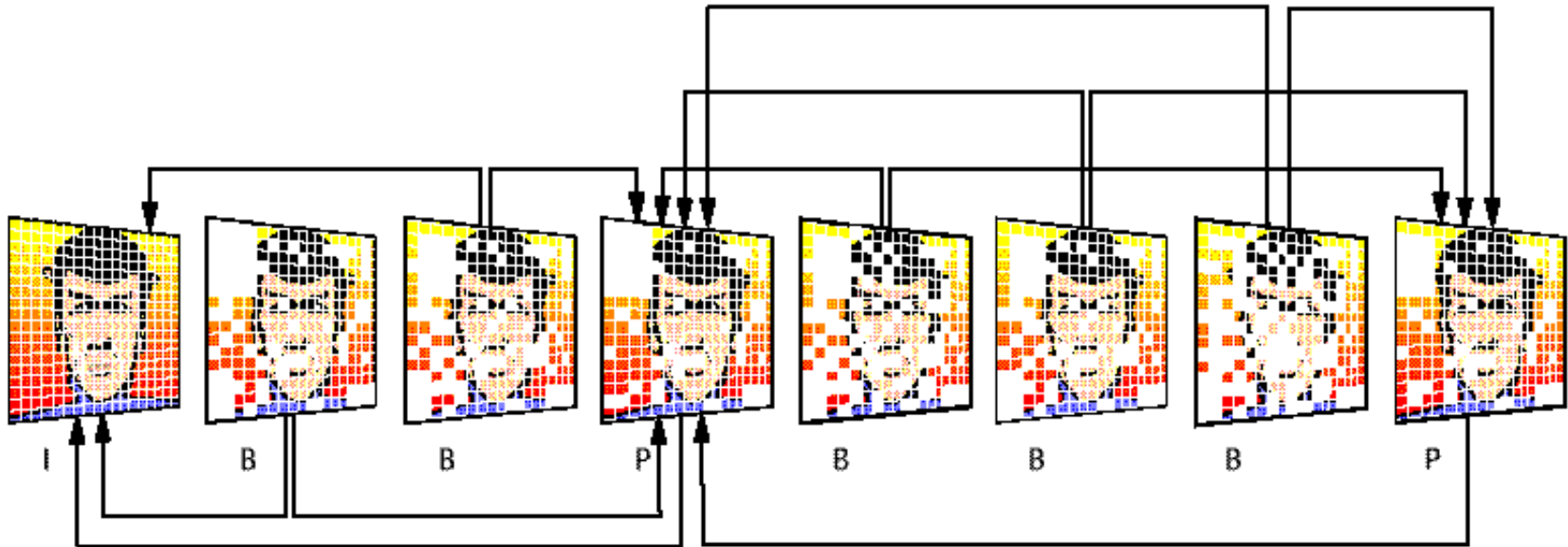


Temporal Redundancy Reduction (motion vectors)



Types of frames in compression

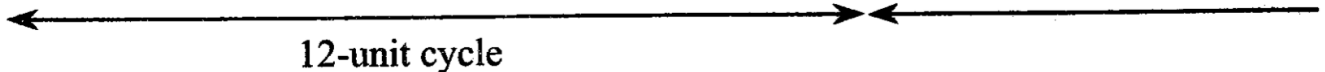
- MPEG uses three types of frames for video coding (compressing)
 - I frames: intra-frame coding
 - Coded without reference to other frames
 - Moderate compression (DCT, JPEG-like)
 - Access points for random access
 - P frames: predictive-coded frames
 - Coded with reference to [previous](#) I or P frames
 - B frames: bi-directionally predictive coded
 - Coded with reference to [previous and future](#) I and P frames
 - Highest compression rates



- *I* frames are independently encoded
- *P* frames are based on previous *I* and *P* frames
- *B* frames are based on previous and following *I* and *P* frames

Sequence of frames

Frame number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	...
Frame type	I	B	B	P	B	B	P	B	B	P	B	B	I	B	B	P	B	B	...



12-unit cycle

(a)

Encoded frame order	1	4	2	3	7	5	6	10	8	9	13	11	12	16	...
Frame type	I	P	B	B	P	B	B	P	B	B	I	B	B	P	...

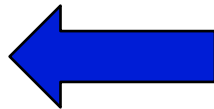
(b)

Type Size Compression

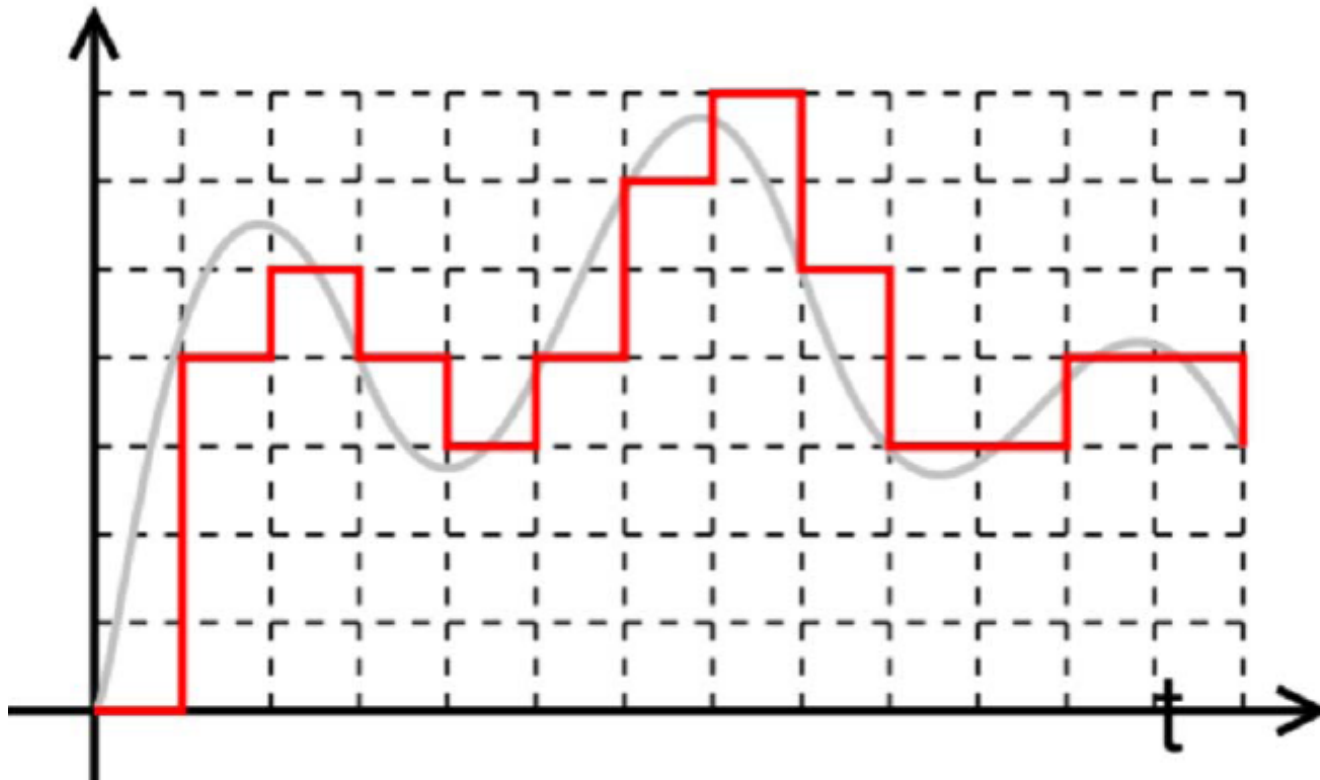
I	18 KB	7 : 1
P	6 KB	20 : 1
B	2.5 KB	50 : 1
Avg	4.8 KB	27 : 1

Representation of information within a computer

- Numbers
- Text (characters and ideograms)
- Documents
- Images
- Video
- Audio



Digitization of audio (analog) signals

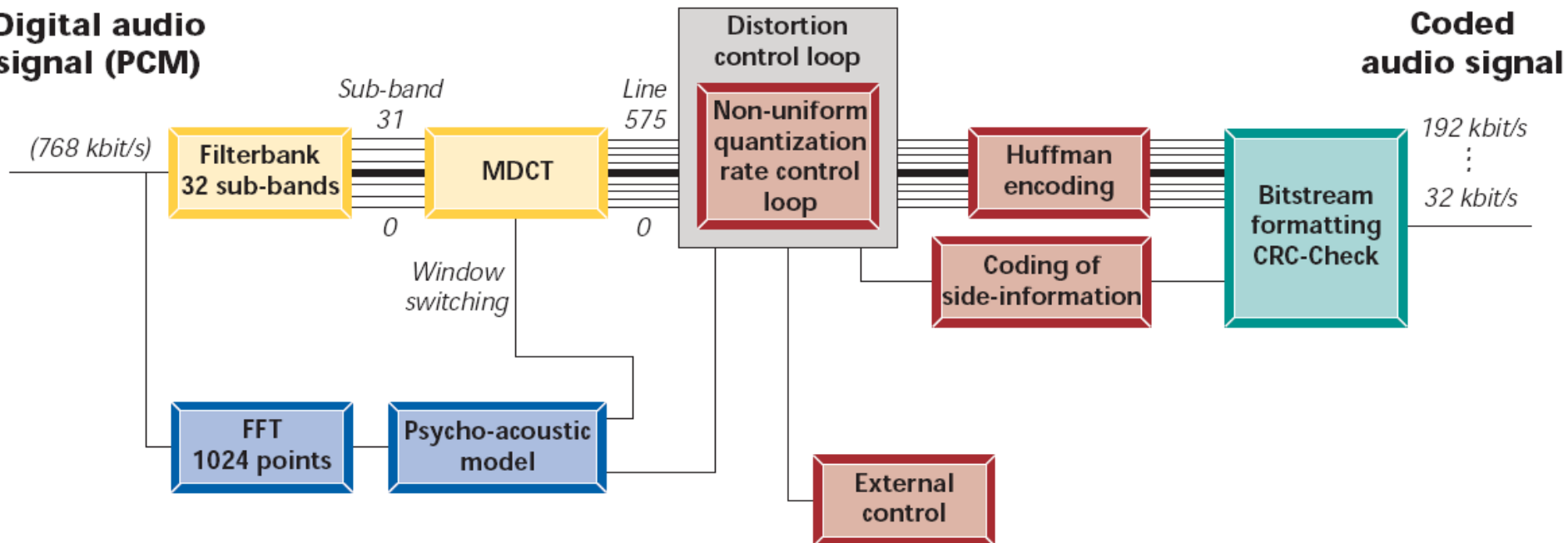


- sampling rate should be at least the double of the highest frequency in the signal (Shannn theorem)
- 8-16 bit per sample

Representing audio

- MPEG-1 defines three different schemes (called *layers*) for compressing audio
- All layers support sampling rates of 32, 44.1 and 48 kHz
- MP3 is MPEG-1 Layer 3

Digital audio signal (PCM)



MPEG summary

- The main aim of MPEG-1 and –2 is to efficiently code compressed video and audio (e.g. MP3 in MPEG-1 and DVD video in MPEG-2)
- The main aim of MPEG-4 is to extend the audio/video stream with additional information and capabilities, such as still images, 3D objects, animation (a la GIF), some interactivity, etc. It contains also further improvements for compression (used in DivX)
- MPEG-1, -2 and –4 have been defined to represent, in a compressed form, the multimedia content (“the bits”)
- MPEG-7 has been defined with a different aim, i.e. to represent information about the multimedia content (it is the “bits about the bits”) and is substantially a metadata set
- MPEG-21 has been defined with the aim of providing a further level of description of the multimedia content, to represent its complete life-cycle and to represent it in a more abstract way, as “Digital Item”

Multimedia file formats

- A **muxer** (abbreviation of multiplexer) is a “container” file that can contain several video and audio streams, compressed with codecs
 - Common file formats are AVI, DIVx, FLV, MKV, MOV, MP4, OGG, VOB, WMV, 3GPP
- A **codec** (abbreviation of coder/decoder) is a “system” (a series of algorithms) to compress video and audio streams
 - Common video codecs are HuffYUV, FLV1, HEVC, Mpeg2, xvid4, x264, H264, H265
 - Common audio codecs are AAC, AC3, MP3, PCM, Vorbis

Refresher on Computer Fundamentals and Data Representation

- Brief History of computers
- Architecture of a computer
- Data representation
- Metadata

