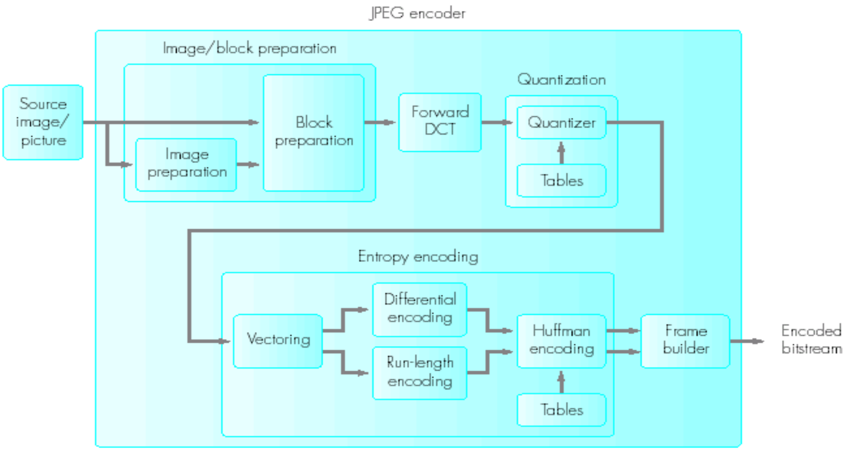
CMR
INSTITUTE OF
TECHNOLOGY

USN | | | | | | | | | |

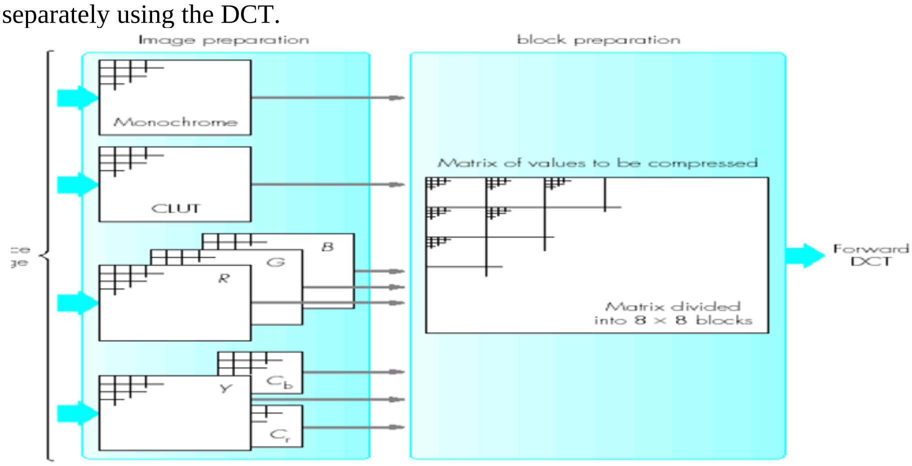
Internal Assesment Test - II

Sub:	Multimedia Communication							Code:	17EC741
Date:	20/12/2021	Duration:	90 mins	Max Marks:	50	Sem:	VII	Branch:	ECE-A,B,C,D
Answer Any FIVE FULL Questions									

		Marks	OBE	
			CO	RBT
1.	Explain the schematic of JPEG encoder with the help of suitable diagrams.	[10]	CO3	L2



- **Source image -** is made up of one or more 2-D matrices of 8-bit grey-level values that represent the image.
- For the colour image if a **CLUT** is used then a single matrix of values is required.
- For **R, G, B** format image - **3 matrices** are required each for R G B.
- For **Y, C_b, C_r** format image- matrix size for the chrominance components is smaller than the Y matrix (Reduced representation).
- **Block preparation -** is carried out before DCT for efficient transformation.
- Once the image format is selected then the values in each matrix are compressed separately using the DCT.

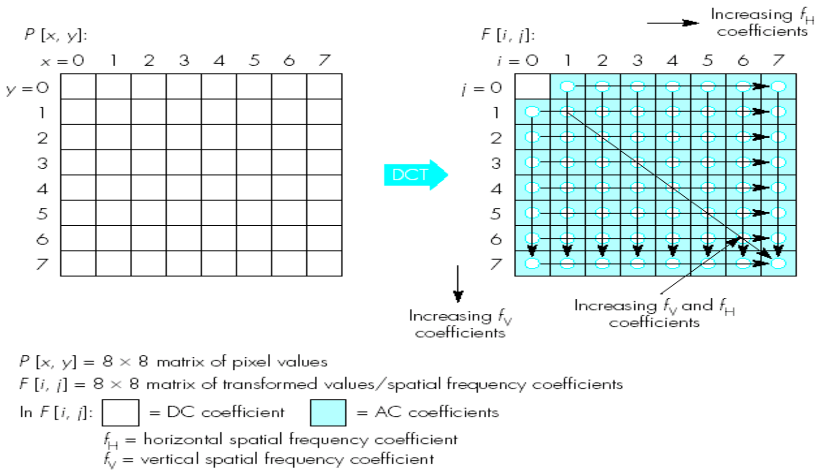


- **Block preparation-** each matrix is divided into a set of smaller 8x8 sub-matrices (block).
- These *BLOCKS* are fed sequentially to the DCT.
- Values in each matrix are compressed separately using the DCT.
- Block preparation is necessary since computing the transformed value for each position in a matrix requires the values in all the locations to be processed.
- Each pixel value is quantized using ***8 bits*** which produces a value in the range ***0 - 255*** for the ***R, G, B*** or ***Y*** and a value in the range ***–128 to +127*** for C_b and C_r.

- To compute forward DCT, all the values are first centred around zero by subtracting ***128*** from each intensity/ Y value.
- If the *input matrix* is ***P[x,y]*** and the *transformed matrix* is ***F[i,j]*** then the DCT for the ***8X8 block*** is computed using the expression:
- ***C(i) & C(j) = ½ for i=j=0 & 1*** for all other values of i,j.

$$F[i,j] = \frac{1}{4} C(i)C(j) \sum_{x=0}^{7} \sum_{y=0}^{7} P[x,y] \cos\frac{(2x+1)i\pi}{16} \cos\frac{(2y+1)j\pi}{16}$$

- All 64 values in the input matrix ***P[x,y]*** contribute to each entry in the transformed matrix ***F[i,j].***
- For ***i=j=0*** the 2 cosine terms are 0 and hence the value in the location ***F[0,0]*** of the transformed matrix is a ***function of the summation of all the values in the input matrix.***
- It is the ***mean of all 64 values*** in the matrix-**DC coefficient.**
- **V**alues in all the other locations of the transformed matrix have a frequency coefficient - **AC coefficients.**
- for ***j=0*** only the horizontal frequency coefficients are present.
- for ***i=0*** only the vertical frequency components are present.
- For all the other locations both the horizontal and vertical frequency coefficients are present.



- Using DCT there is *very little loss of information* during the DCT phase.
- The losses are due to the use of fixed point arithmetic.
- The main source of information loss occurs during the *quantization and entropy encoding* stages where the compression takes place.
- The ***human eye responds primarily to the DC coefficient*** and the lower frequency coefficients (*The higher frequency coefficients below a certain threshold will not be detected by the human eye*).
- **Quantization** - *reduce the size of the DC and AC coefficients* - less bandwidth is required for their transmission -by using a divisor.
- Human *eye sensitivity varies* with spatial frequency -amplitude threshold below which the eye will detect a particular frequency also varies.
- Threshold values vary for each of the 64 DCT coefficients and are held in a 2-D matrix known as the **quantization table.**
- JPEG has *two quantization tables* - ***one for luminance*** and other for ***chrominance coefficients***. Customized tables are also allowed to be sent with the compressed image.

DCT coefficients

120	60	40	30	4	3	0	0
70	48	32	3	4	1	0	0
50	36	4	4	2	0	0	0
40	4	5	1	1	0	0	0
5	4	0	0	0	0	0	0
3	2	0	0	0	0	0	0
1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0

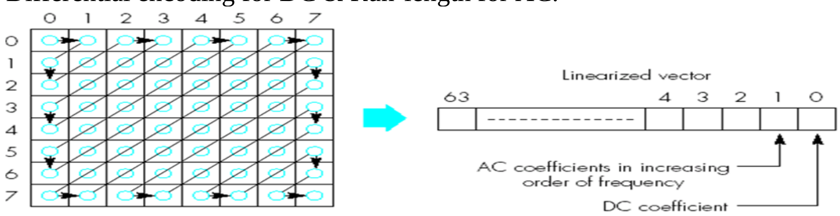
Quantizer

Quantized coefficients

12	6	3	2	0	0	0
7	3	2	0	0	0	0
3	2	0	0	0	0	0
2	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

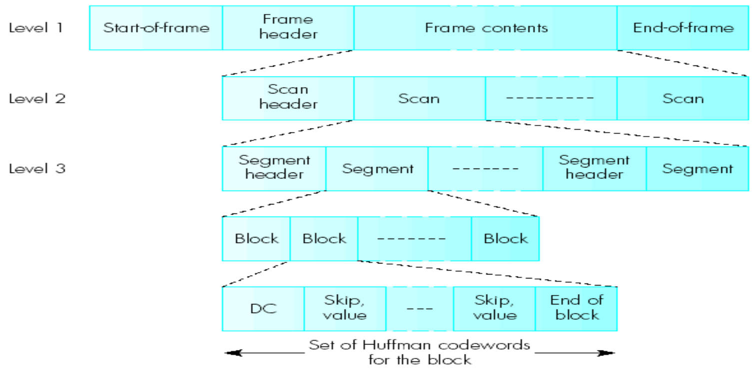
10	10	15	20	25	30	35	40
10	15	20	25	30	35	40	50
15	.20	25	30	35	40	50	60
20	25	30	35	40	50	60	70
25	30	35	40	50	60	70	80
30	35	40	50	60	70	80	90
35	40	50	60	70	80	90	100
40	50	60	70	80	90	100	110

Quantization table

- From the *quantization table* and the *DCT and quantization coefficients* number of observations can be made:
    - The computation of the quantized coefficients involves *rounding the quotients* to the nearest integer value.
    - The threshold values used increase in magnitude with increasing spatial frequency.
    - The DC coefficient in the transformed matrix is largest.
    - Many of the higher frequency coefficients are zero.
    - 4 stages – *vectoring, differential, run-length, huffman encoding.*
    - ***Vectoring*** – entropy encoding operates on a 1D string of values (vector). But output of the quantization is a 2D matrix - hence this has to be represented in a 1D form.
    - **Zig-zag scan** is used to remove large no. of 0.
    - DC and lower frequency AC components are scanned first.
    - Differential encoding for DC & Run-length for AC.
    - 

    - ***Differential encoding*** – difference in magnitude of the DC coefficient in a quantized block relative to the value in the preceding block is encoded – it reduce the number of bits required to encode the relatively large DC magnitude.
    - *E*.g: DC coefficients in consecutive quantized blocks was: 12, 13, 11, 11, 10, - difference values will be 12, 1, -2, 0, -1.
    - The difference values are encoded in the form *(SSS, value) – SSS=no. of bits needed, value=bits that represent the value.*
    - Positive value – unsigned binary, negative value – compliment of it.
    - The remaining 63 values in the vector are the AC coefficients.
    - Due to large number of 0's in the AC coefficients they are encoded as ***string of pairs of values.***
    - **Each pair** - (*skip, value*); *skip* = no. of zeros in the run and *value* = next non-zero coefficient.
    - The above will be encoded as:*(0,6) (0,7) (0,3) (0,3) (0,3) (0,2)(0,2)(0,2)(0,2)(0,0)*
    - **Final pair *(0,0)*** - indicates the end of the string for this block.
    - JPEG include a definition of the structure of the total bitstream relating to a particular image/picture - ***frame.***
    - ***Frame builder*** - *encapsulate* all the information relating to an encoded image/picture. Frame has hierarchical structure.

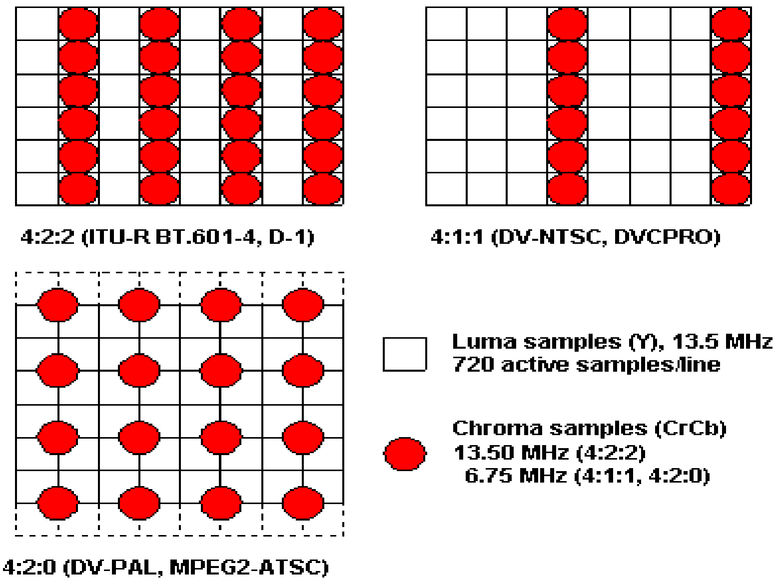
- ***At the top level*** - complete ***frame-plus-header is encapsulated*** between a *start-of-frame* and an *end-of-frame delimiter* –receiver determines the start and end of all the information relating to a complete image.



- Frame header contains a number of fields:
- overall width and height of the image in pixels.
- no. and type of components (CLUT, R/G/B,Y/C_b/C_r).
- digitization format used (4:2:2, 4:2:0 etc.).
- At the next level a frame consists of a no. of components – each is known as a *scan, with header containing fields :*
- identity of the components.
- no. of bits used to digitize each component.
- quantization table used to encode each component.
- Each *scan* comprises one or more *segments* each of which can contain a group of (8X8) *blocks* preceded by a header.
- This contains the set of Huffman codewords for each block.

2. Distinguish between 4:2:2 and 4:2:0 digitization formats. [10] CO2 L1

- **Digitization format :** ***4:2:2, 4:2:0, HDTV, SIF, CIF, QCIF***
- **4 : 2 : 2 format** -original digitization format used in CCIR. Used in TV studio.
- Y bandwidth – ***6Mhz***, Cb n Cr bandwidth – ***3Mhz***.
- Line sampling rate for ***Y=13.5MHz*** for ***Cb*** & ***Cr = 6.75MHz***
- ***625 line system – line sweep time= 64µs -12 µs retrace = 52 µs.***
- The ***no. of Y samples per line = 52 µ x 13.5M = 720 samples/line.***
- No. of samples for each of the two chrominance signals ***-360 samples per active line.***
- This results in ***4Y*** samples for every ***2Cb***, and ***2Cr*** samples.



- 4 : 2 : 0 format is used in digital *video broadcast applications.*
- **Interlaced scanning** is used and the absence of chrominance samples in alternative

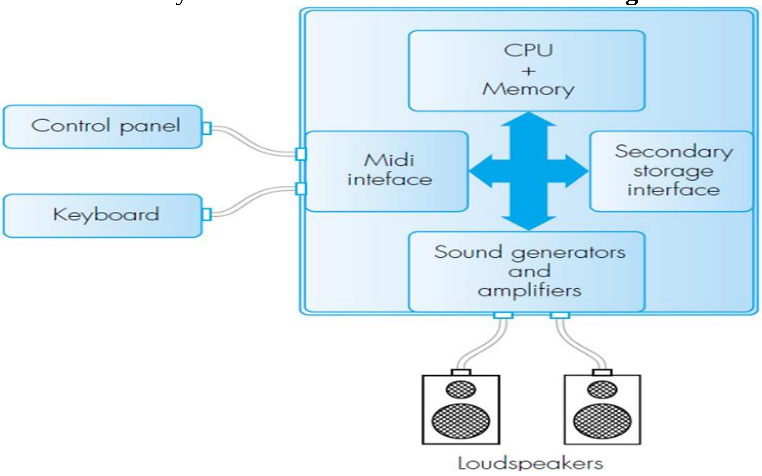
lines.

- The same luminance resolution but half the chrominance resolution.
- Bit rate in both system = **Y BW x 8 bits + 2 x (Cb/Cr BW x 8 bits)**
- 525line system – Y = 720 x 480
  Cb = Cr =360 x 240
- 625 line system - Y = 720 x 576
  Cb = Cr =360 x 288
- Frame refresh rate – 25/30Hz.

$$13.5\times10^{6}\times8+2(3.375\times10^{6}\times8)=162Mbps$$

3a).	Explain the operation of audio synthesizer, with suitable diagram.	[5]	CO2	L1

- Digitized audio is easy to store in computer, but large memory needed.
- **Synthesized audio** is used for multimedia application - uses **less memory** than digitized audio. [5]
- It is easier to edit synthesized audio and mix several passages together.
- 3 main components - computer, keyboard, sound generators.
- *Computer* takes input from *Keyboard* sends commands to *sound generators* which produces Sound waveform - to drive speakers.
- Each key has a different codeword – called **message** that is read by the computer.



- ***Control panel*** - switches & sliders – volume control & sound effects.
- Secondary storage interface – for storing audio in secondary Storage devices – floppy disk.
- Different programs - to edit a previously entered passage or mix several stored passages together.
- **Music instrument digital interface-Midi –** defines formats for inputs from different instruments, types of connectors, cables, electrical signals- 16 channels 1 for each instrument.
- *Midi message format*- consists of :
  - ***Status byte*** – defines particular event that has caused the message to be generated – eg = key pressed on keyboard.
  - ***Data byte*** – no. of data bytes defines parameters for the event. – eg = key identity and pressure applied on key.

b) Derive the bit rate and the memory requirements to store each frame that result from the digitization of both a 525-line and a 625-Line system assuming a 4:2:2 format. Also find the total memory required to store a 1.5 hour movie/video.

525-line system: The number of samples per line is 720 and the number of visible lines is 480. Hence the resolution of the luminance (Y) and two chrominance (C_b and C_r) signals are:

$$Y = 720 \times 480$$
$$C_b = C_r = 360 \times 480$$

Bit rate: Line sampling rate is fixed at 13.5 MHz for Y and 6.75 MHz for both C_b and C_r, all with 8 bits per sample.

Hence: Bit rate $= 13.5 \times 10^6 \times 8 + 2\ (6.75 \times 10^6 \times 8) = 216$ Mbps

Memory required: Memory required per line $= 720 \times 8 + 2\ (360 \times 8)$
$= 11\,520$ bits or 1440 bytes

Hence memory per frame, each of 480 lines $= 480 \times 11\,520$
$= 5.5296$ Mbits or 691.2 kbytes

and memory to store 1.5 hours assuming 60 frames per second:
$= 691.2 \times 60 \times 1.5 \times 3600$ kbytes
$= 223.9488$ Gbytes

625-line system: Resolution: $Y = 720 \times 576$
$C_b = C_r = 360 \times 576$

Bit rate $= 13.5 \times 10^6 \times 8 + 2\ (6.75 \times 10^6 \times 8) = 216$ Mbps

Memory per frame $= 576 \times 11\,520 = 6.635\,55$ Mbits or 829.44 kbytes

and memory to store 1.5 hours assuming 50 frames per second:
$= 829.44 \times 50 \times 1.5 \times 3600$ kbytes
$= 223.9488$ Gbytes

4. Messages comprising seven different characters, A through G, are to be transmitted over a data link. Analysis has shown that the relative frequency of occurrence of each character is A 0.10, B 0.25, C 0.05, D 0.32, E 0.01, F 0.07, G 0.2 (i) Derive the entropy of the messages (ii) Use static Huffman coding to derive a suitable set of codeword and construct a tree (iii) Derive the average number of bits per codeword for your codeword set. [10] CO3 L3

(c) Derive the average number of bits per character for your codeword set and compare this with:
 (i) the entropy of the messages (Shannon's value),
 (ii) fixed-length binary codewords,
 (iii) 7-bit ASCII codewords.

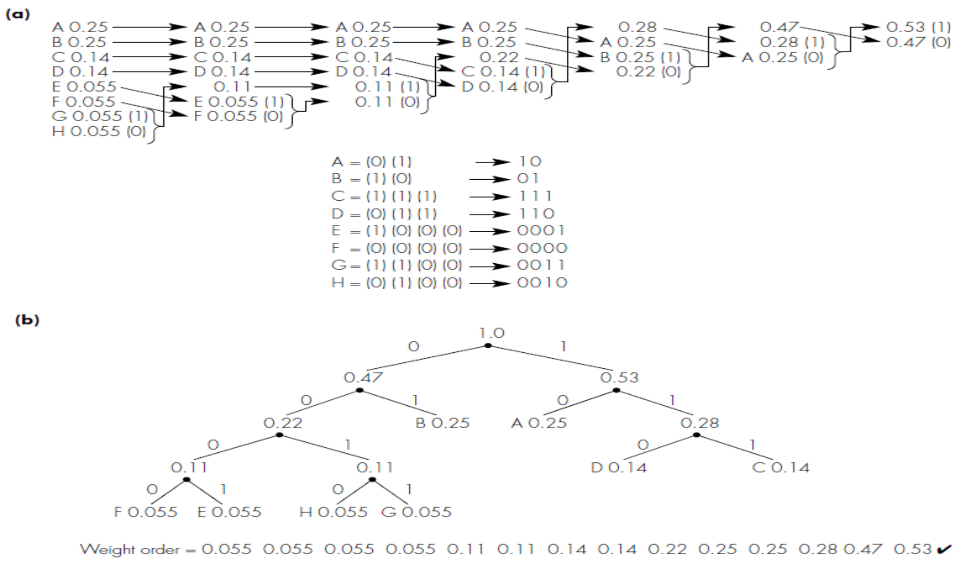
Answer:

(a) Shannon's formula states:

$$\text{Entropy, } H = -\sum_{i=1}^{8} P_i \log_2 P_i \text{ bits per codeword}$$

Therefore:

$$H = -(2(0.25 \log_2 0.25) + 2(0.14 \log_2 0.14) + 4(0.055 \log_2 0.055))$$
$$= 1 + 0.794 + 0.921 = 2.175 \text{ bits per codeword}$$



(b) The derivation of the codeword set using Huffman coding is shown in Figure 3.4(a). The characters are first listed in weight order and the two characters at the bottom of the list are assigned to the (1) and (0) branches. Note that in this case, however, when the two nodes are combined, the weight of the resulting branch node (0.11) is greater than the weight of the two characters E and F (0.055). Hence the branch node is inserted into the second list higher than both of these. The same procedure then repeats until there are only two entries in the list remaining.

The Huffman code tree corresponding to the derived set of codewords is given in Figure 3.4(b) and, as we can see, this is the optimum tree since all leaf and branch nodes increment in numerical order.

(c) Average number of bits per codeword using Huffman coding is:

$2\,(2 \times 0.25) + 2(3 \times 0.14) + 4(4 \times 0.055) = 2.72$ bits per codeword
which is 99.8% of the Shannon value.

Using fixed-length binary codewords:

There are 8 characters – A through H – and hence 3 bits per codeword is sufficient which is 90.7% of the Huffman value.

Using 7-bit ASCII codewords:

7 bits per codeword
which is 38.86% of the Huffman value.

5. What is the need for Compression? Explain briefly about entropy encoding. [10] CO3 L2

- **Compression** is done prior to transmission
  - to *reduce volume* of data - text & image [10] CO3 L3
  - to *reduce transmission bandwidth* – audio & videos
- **Reduction in data size** – reduces required *bandwidth*, *storage* and call *cost.*
- Source encoder – compression & destination decoder – decompression. [10] CO2 L1

- **Entropy** – lossless – independent of the type of data that is being compressed – concerned with how data is represented.
- **1) *Run-length encoding* -** are when the source data comprises *long substrings* of the same characters/bits.
- Source string -transmitted as a different set of codewords - indicating **character** and **no. of bits** in the substring.
- Destination - *interprets* each codeword received and outputs the appropriate no. of characters/bits. – *provided destination knows the set of codewords being used.*
- E.g. – scanner output in a fax machine - *long substring*

a) 000000011111111110000011 - represented as
0,7 1,10 0,5 1,2

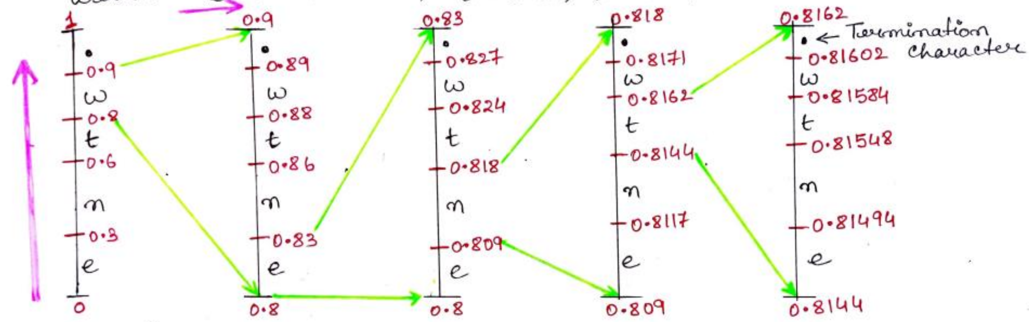
b) aaaaabbbbbbccccyy – **5a6b4c2y**
5 a 6 b 4c 2 y

- RLE - technique used to reduce the size of a **repeating string of characters** called a ***run***.
- RLE encodes a run of symbols into ***2 bytes-count & symbol.***
- RLE can compress any type of data regardless of its information content, but the content of data to be compressed affects the compression ratio.
- RLE - *low compression ratios* than other compression methods – but easy to implement and quick execution.
- Supported by most bitmap file formats -TIFF, BMP, PCX.
- Compression is normally measured with the **compression ratio** = ***original size / compressed size*** : 1
- **2) *Statistical encoding*** -Strings are transmitted using ASCII codeword. Symbol occurrence is variable in the source data.
- *E.g* - ***A*** may occur more frequently than ***P.***
- Statistical coding uses this property - by using **variable length codeword** – *shortest codeword used for representing the most frequently appearing symbol*.

- **Entropy of source** – *minimum average no. of bits required to transmit source data* - **Shannon Entropy H**
- ***n***= no. of diff. symbols. ***Pi*** = probability of occurrence of symbol ***i***. ***Avg. no. of bits/codeword = (ΣNiPi).***
- ***Encoding efficiency=Source Entropy/avg. no. of bits.***

6. Explain the concept of Arithmetic coding. Compute the arithmetic code word for the message "went." Comprising a string of characters with probabilities e=0.3, n=0.3, t=0.2, w=0.1, **.** =0.1

went. e = 0.3, n = 0.3, t = 0.2, w = 0.1, • = 0.1.



Calculation

→ d = 0.9 − 0.8 = 0.1 for expanding w.

Range of e = 0.8 : 0.8 + 0.1 X 0.3 = 0.8 : 0.83

Range of n = 0.83 : 0.83 + 0.1 X 0.3 = 0.83 : 0.86

Range of t = 0.86 : 0.86 + 0.1 X 0.2 = 0.86 : 0.88

Range of w = 0.88 : 0.88 + 0.1 X 0.1 = 0.88 : 0.89

Range of • = 0.89 : 0.9

→ Message went. is converted into arithmatic codeword.

→ Range of the codeword

0.81602 < codeword < 0.8162

→ d = 0.83 − 0.8 = 0.03 for expanding e

Range of e = 0.8 : 0.8 + 0.03 X 0.3 = 0.8 : 0.809

Range of n = 0.809 : 0.809 + 0.03 X 0.3 = 0.809 : 0.818

Range of t = 0.818 : 0.818 + 0.03 X 0.2 = 0.818 : 0.824.

Range of w = 0.824 : 0.824 + 0.03 X 0.1 = 0.824 : 0.827

Range of • = 0.827 : 0.83

→ d = 0.818 − 0.809 = 0.009 for expanding n

Range of e = 0.809 : 0.809 + 0.009 X 0.3 = 0.809 : 0.8117

Range of n = 0.8117 : 0.8117 + 0.009 X 0.3 = 0.8117 : 0.8144

Range of t = 0.8144 : 0.8144 + 0.009 X 0.2 = 0.8144 : 0.8162 .

Range of w = 0.8162 : 0.8162 + 0.009 X 0.1 = 0.8162 : 0.8171.

Range of • = 0.8171 : 0.818

→ d = 0.8162 − 0.8144 = 0.0018 for expanding t

Range of e = 0.8144 : 0.8144 + 0.0018 X 0.3 = 0.8144 : 0.8194

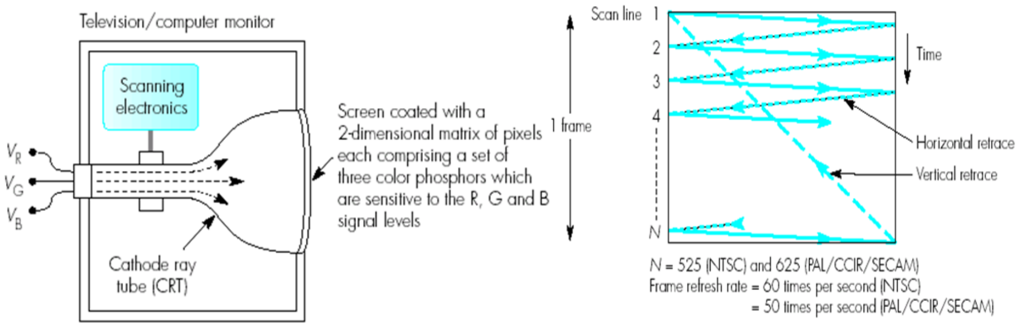
Range of n = 0.8194 : 0.8194 + 0.0018 X 0.3 = 0.8194 : 0.81548

Range of t = 0.81548 : 0.81548 + 0.0018 X 0.2 = 0.81548 : 0.81584

Range of w = 0.81584 : 0.81584 + 0.0018 X 0.1 = 0.81584 : 0.81602.

Range of • = 0.81602 : 0.8162.

7. Explain briefly about Raster scanning and Interlaced Scanning Procedure.



- ***Finely-focussed electron beam*** being scanned over the complete screen.
- Each complete scan has no. of discrete horizontal lines - *starts at the top left corner* and *ends at the bottom right corner*, then the beam is deflected back again to the top left corner – *retrace path*: beam is off.
- **Frame -** each complete set of horizontal scan.
- Each frame is made of ***N individual scan lines***: N= ***525 – N & S America*** & ***625 – Europe and other countries***.
- Inside of display screen is coated with ***Light-sensitive phosphor*** – that emits light when energized by electron bean.
- Amount of light emitted- **brightness** is determined by the *power* in the electron beam.
- ***Black and white picture tube*** – *single electron beam* with white-sensitive phosphor.
- ***Color tube*** - 3 separate closely located beams and a 2D matrix of pixels.
- Each pixel comprises of a set of 3 related *colour-sensitive phospors* associated with *each pixel* is called a ***phospor triad.*** **RED GREEN BLUE.**
- The light/color produced by the phosphor is designed to *decay very quickly* – continuous ***screen refreshment*** is needed.
- **Scanning sequence** - It is necessary to use a minimum refresh rate of 50 times per second to avoid flicker.
- A refresh rate of ***25 times per*** second is sufficient.
- **Field-** image is divided into odd scan lines and even scan lines – each are called *fields.*
- **Interlaced scanning –** each field is scanned separately and integrated in the television receiver.
- Used in ***transmission***, due to lower
-    frame rate.
- ***Image*** – representation in RGB,
-    raster scan.
- ***Video*** –representation in ***YCbCr***
-    interlaced scan.

Figure 2.19 Interlaced scanning principles.