Hamming Code

A **Hamming code** is a linear error-correcting code named after its inventor, Richard Hamming. Hamming codes can detect up to two bit errors, and correct single-bit errors. This method of error correction is best suited for situations in which randomly occurring errors are likely, not for errors that come in bursts.

- Course Name: Error Correcting Codes Level(UG/PG): UG
- Author: Phani Swathi Chitta
- Mentor: Prof. Saravanan Vijayakumaran

Learning Objectives

After interacting with this Learning Object, the learner will be able to:

Explain the

- Encoding of message bits to create transmitted codeword
- Decoding of received codeword using Hamming code

Definitions of the components/Keywords:

- For a Hamming code, the minimum distance is exactly 3.
  - Hence, the code is capable of correcting all the error patterns with a single error or detecting all the error patterns of two or fewer errors.

- A Single Parity Check (SPC) code is a linear block code with a single parity check digit which can detect single bit errors.
  - The parity bit is appended to the information bits and is set to 1 if the number of ones in the information bits is odd and is set to 0 if the number of ones in the information bits is even. Thus the resultant codeword which consists of the information bits and the parity bit will have an even number of ones.
  - An even parity check which involves taking modulo 2 sum of all the received bits and checking if it zero can detect single bit errors.

- Hamming code extends this by using multiple even parity checks to correct single bit errors.
  - To correct a single bit error it is sufficient to know the location of the error since correction involves flipping the bit at the error location
  - In the Hamming code, we conduct multiple even parity checks and for each one of them we output 1 if they fail and 0 if they pass
  - We want the sequence of 1's and 0's to form the binary representation of the error location in the received vector.

Definitions of the components/Keywords:

- For a (7,4) Hamming code,
  - the first even parity check should involve all the odd numbered locations 1,3,5,7 because these locations have a 1 in the least significant bit of their binary representations
  - the second even parity check has to involve locations 2,3,6,7 because these locations have a 1 in the next to least significant bit of their binary representations
  - the third even parity check has to involve locations 4,5,6,7
  - at least one of the bit in each set of locations is a parity bit which will be 0 or 1 in order to make the number of ones in the locations even.

- Here 1,2,4 are the parity bits and 3,5,6,7 are the information bits

1	2	3	4	5	6	7
P_1	P_2	U_3	P_4	U_5	U_6	U_7

If a single error happens in a Hamming Code, the sequence of failed and passed even parity checks or **syndrome** gives the binary representation of the location of the single bit error. If syndrome is all zeros then assume that no error occurred

Definitions of the components/Keywords:

1

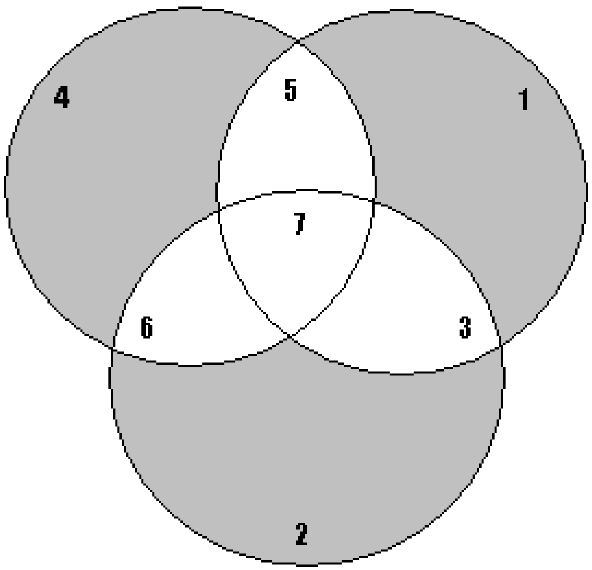
2

3

4

5

- The Venn diagram representation of Hamming Code is,



$$V_1 = u_3 + u_5 + u_7$$
$$V_2 = u_3 + u_6 + u_7$$
$$V_3 = u_3$$
$$V_4 = u_5 + u_6 + u_7$$
$$V_5 = u_5$$
$$V_6 = u_6$$
$$V_7 = u_7$$

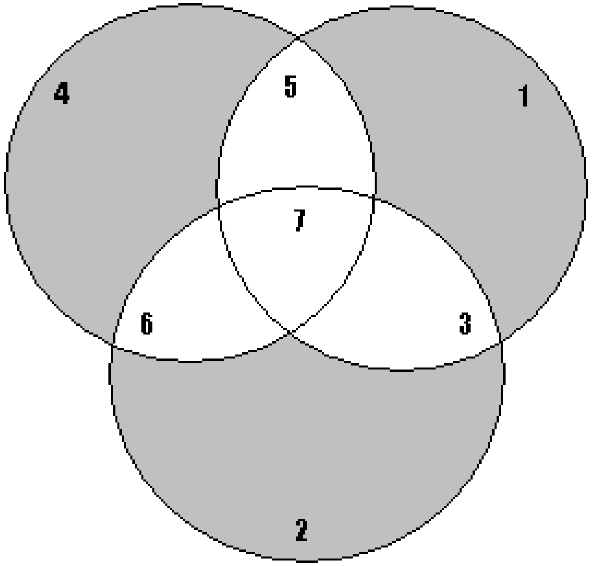
- Where V = (V_1, V_2, V_3, V_4, V_5, V_6, V_7) is the codeword

Master Layout

1
2
3
4
5

Enter the information bits :

3	5	6	7



Encoder

Step 1:

For a (7,4) linear block code, the parity check matrix is

$$H = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Instruction for the animator	Text to be displayed in the working area (DT)
• Show the above statement and matrix	• H is the parity check matrix

Step 2:

The transpose of the parity check matrix is

$$H^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Instruction for the animator	Text to be displayed in the working area (DT)
• Show the above statement and matrix	• The transpose of H matrix

Step 3:

We know $\boldsymbol{r}.\boldsymbol{H}^T = \boldsymbol{0}$ for any codeword

$$\begin{bmatrix} r_1 & r_2 & r_3 & r_4 & r_5 & r_6 & r_7 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = 0$$

Instruction for the animator	Text to be displayed in the working area (DT)
• Show the above statement and matrices	• For any linear block code, we know $\boldsymbol{r}.\boldsymbol{H}^T = \boldsymbol{0}$

Step 4:

1

2

3

4

5

$$r_1 + r_3 + r_5 + r_7 = 0$$
$$r_2 + r_3 + r_6 + r_7 = 0$$
$$r_4 + r_5 + r_6 + r_7 = 0$$

Here 3,5,6,7 are considered as information bits and 1, 2,4 are parity bits

Therefore,

$$r_1 = r_3 + r_5 + r_7$$
$$r_2 = r_3 + r_6 + r_7$$
$$r_4 = r_5 + r_6 + r_7$$

Since addition and subtraction are same in modulo- 2 addition

Instruction for the animator	Text to be displayed in the working area (DT)
• Show the above equations	

Step 5:

Enter the information bits :

1 [3]	0 [5]	0 [6]	1 [7]

Instruction for the animator	Text to be displayed in the working area (DT)
• Show the text above	

Step 6:

Calculating the parity bits:

1	2	3	4	5	6	7
		1		0	0	1

Instruction for the animator	Text to be displayed in the working area (DT)
• Show the above text	• Calculation of parity bits

1

2

3

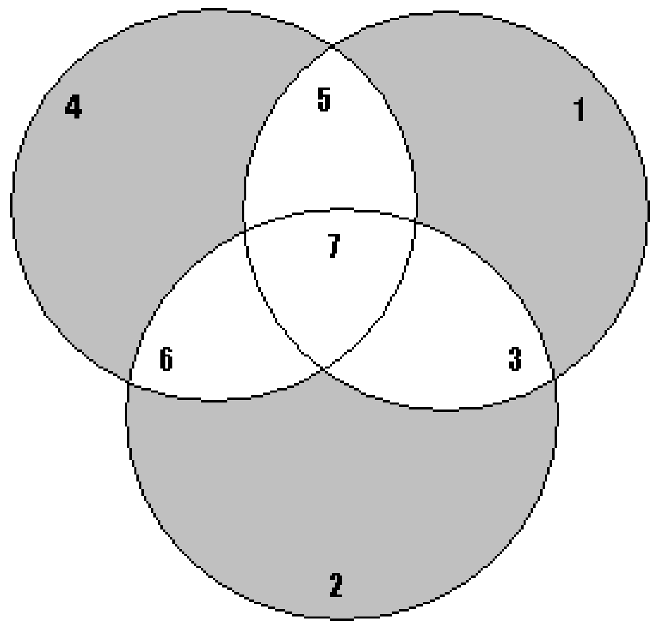
4

5

Step 7: Encoder

1	2	3	4	5	6	7
		1		0	0	1

Venn Diagram for a Hamming Code



Encoder

Instruction for the animator	Text to be displayed in the working area (DT)
• Show the above text and retain the rectangular box till the encoding part	• Calculation of parity bits using Venn diagram

1

2

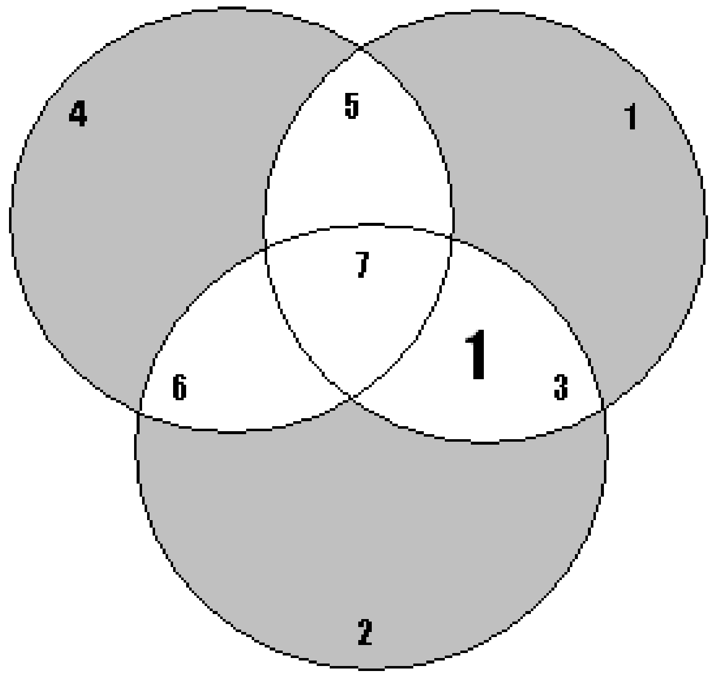
3

4

5

Step 8:

1	2	3	4	5	6	7
		1		0	0	1



Encoder

Instruction for the animator	Text to be displayed in the working area (DT)
• Show the above figure • The bits should appear one by one in a sequence 3,5, 6,7 • It should be shown such that the bits are moving from the rectangular box	• Uploading the information bits into the Venn diagram

1

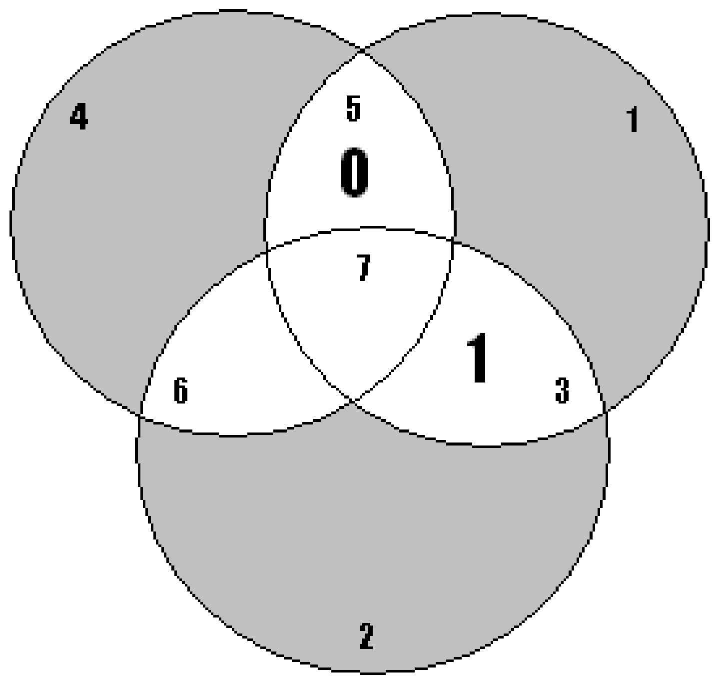
2

3

4

5

Step 9:



Encoder

Instruction for the animator	Text to be displayed in the working area (DT)
• Show the above figure	• Uploading the information bits into the Venn diagram

1

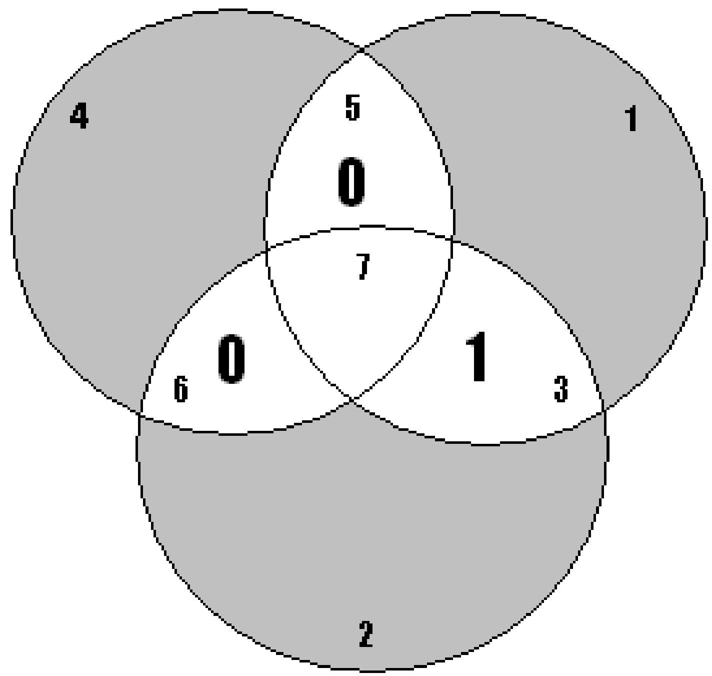
2

3

4

5

Step 10:



Encoder

Instruction for the animator	Text to be displayed in the working area (DT)
• Show the above figure	• Uploading the information bits into the Venn diagram

1

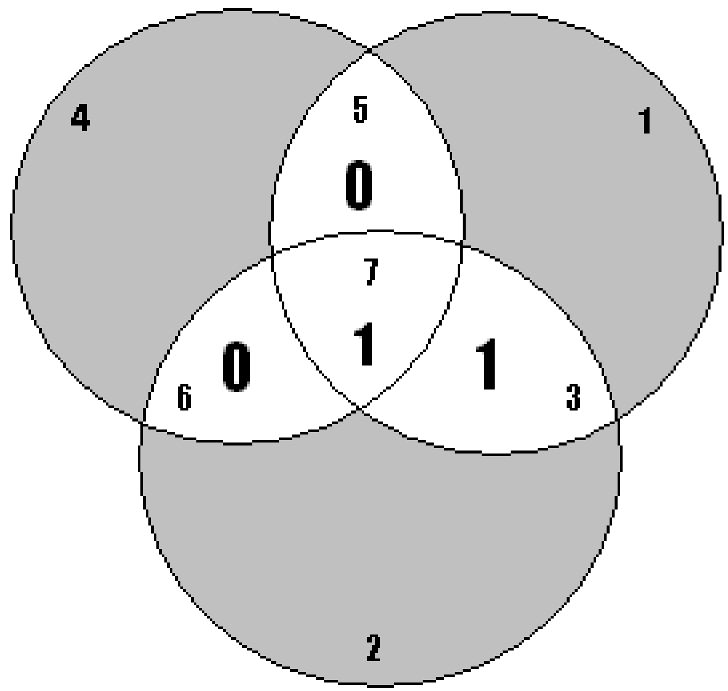
2

3

4

5

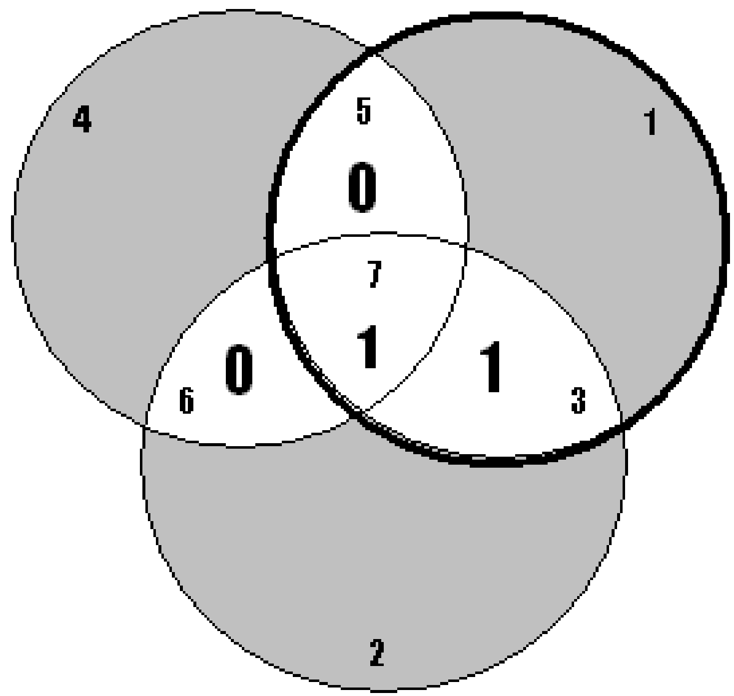
Step 11:



Encoder

Instruction for the animator	Text to be displayed in the working area (DT)
• Show the above figure	• Uploading the information bits into the Venn diagram

Step 12:



Encoder

Instruction for the animator	Text to be displayed in the working area (DT)
• Show the figure as above	• Parity bit for the first set is calculated

1

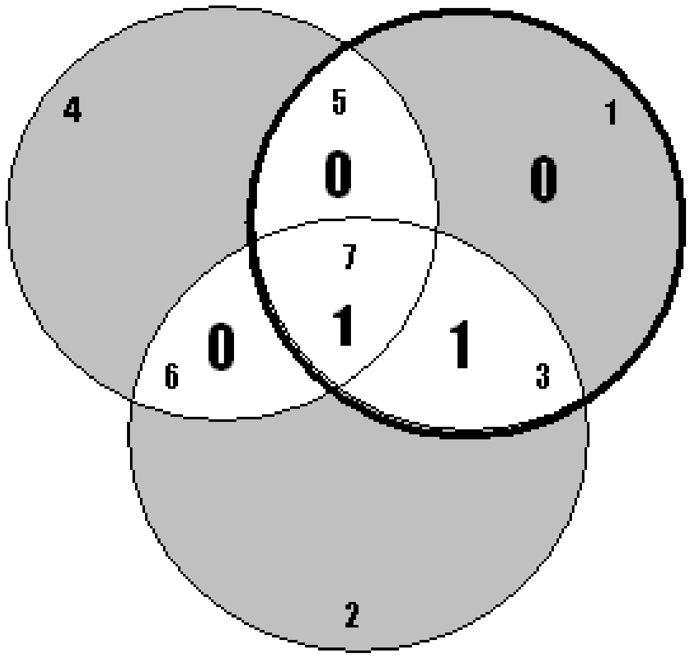
2

3

4

5

Step 13:



Encoder

Instruction for the animator	Text to be displayed in the working area (DT)
• After the dark circle appears, the zero should appear	• The operation is modulo -2 addition $r_1 = r_3 + r_5 + r_7$

1

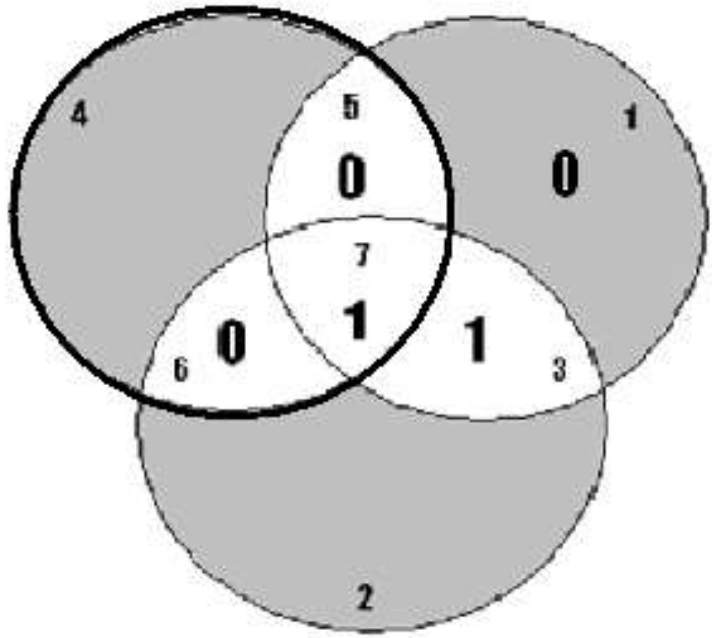
2

3

4

5

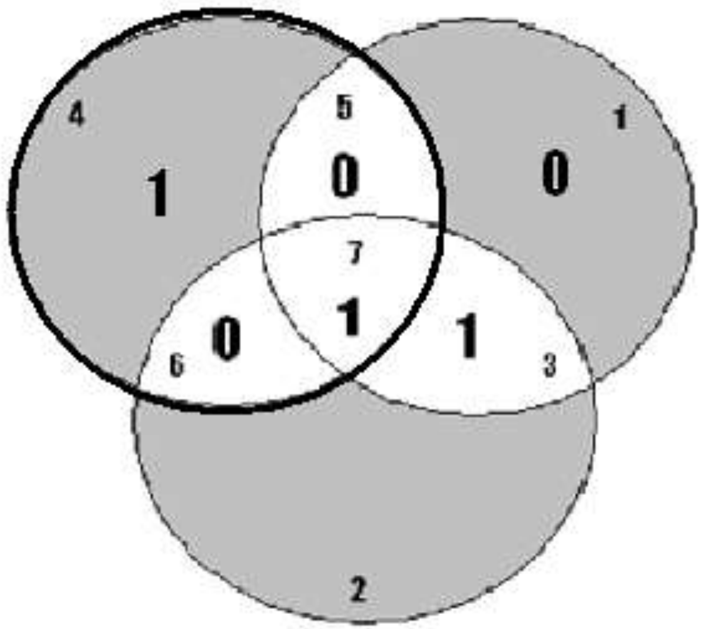
Step 14:



Encoder

Instruction for the animator	Text to be displayed in the working area (DT)
• Show the figure as above	• Parity bit for the second set is calculated

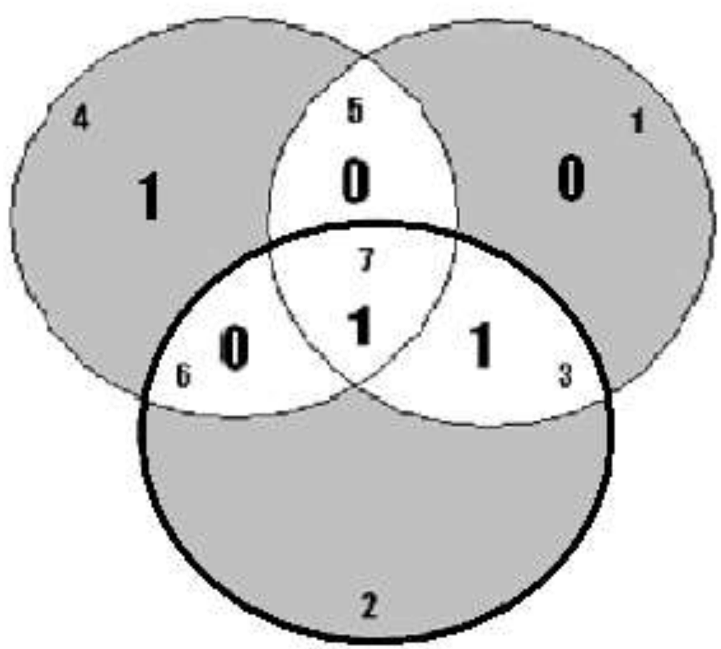
Step 15:



Encoder

Instruction for the animator	Text to be displayed in the working area (DT)
• After the dark circle appears, the one should appear	• The operation is modulo -2 addition $$r_4 = r_5 + r_6 + r_7$$

Step 16:

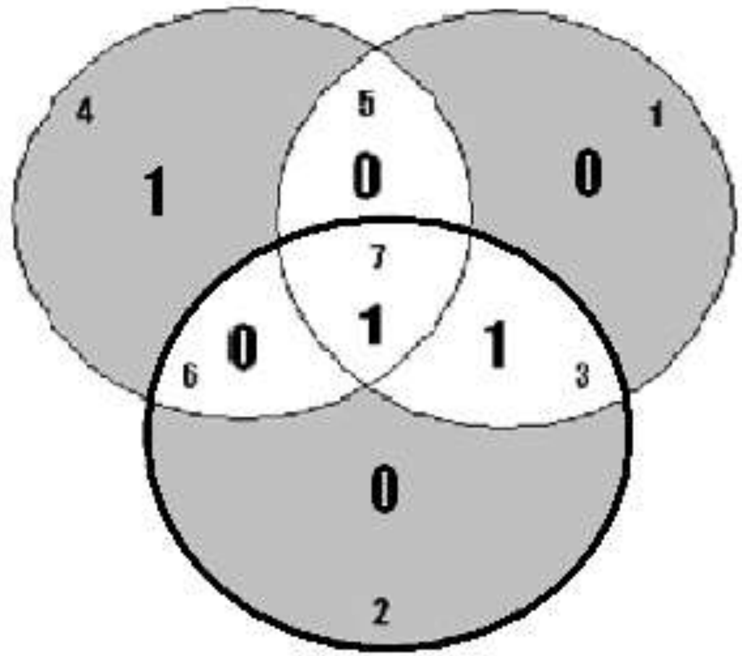


Encoder

Instruction for the animator	Text to be displayed in the working area (DT)
• Show the figure as above	• Parity bit for the third set is calculated

Step 17:

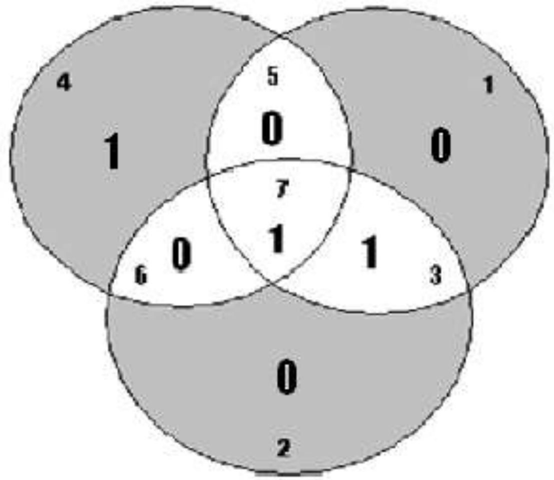
1
2
3
4
5



Encoder

Instruction for the animator	Text to be displayed in the working area (DT)
• After the dark circle appears, the zero should appear	• The operation is modulo -2 addition $r_2 = r_3 + r_6 + r_7$

Step 18:



1	2	3	4	5	6	7
0	0	1	1	0	0	1

Parity Computed

Transmitted code word is: 0 0 1 1 0 0 1

Instruction for the animator	Text to be displayed in the working area (DT)
• Show the circles along with the digits and then show the rectangle with numbers in white boxes. • Then show 0 in box1 , 0 in box2 and 1 in box4 • Then the circles should disappear and the rectangle box must be there along with the transmitted codeword	• Encoding is completed

1

2

3

4

5

Step 19:

Suppose there occurred an error at bit 2, then the received word is:

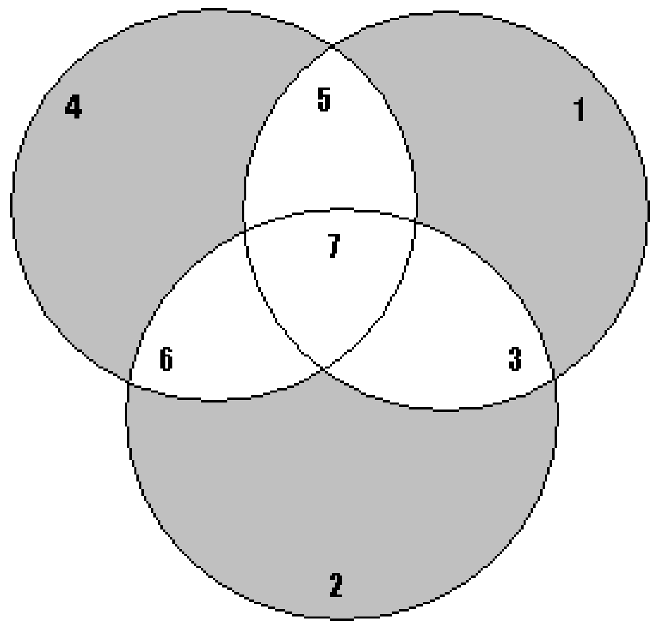
1	2	3	4	5	6	7
0	1	1	1	0	0	1

Received codeword is: 0 1 1 1 0 0 1

Instruction for the animator	Text to be displayed in the working area (DT)
• The rectangular box and the transmitted codeword are retained from the previous slide. • After the transmitted codeword, the first sentence should appear • Then the rectangle box with 1s and 0s (1 in box 2 must be in red) • Then the last sentence should appear	

Step 20: Decoder

0 [1]	1 [2]	1 [3]	1 [4]	0 [5]	0 [6]	1 [7]



Decoder

Instruction for the animator	Text to be displayed in the working area (DT)
• Clear the screen except the rectangular box • Then show the circles	• Decoding using the Venn diagram

1

2

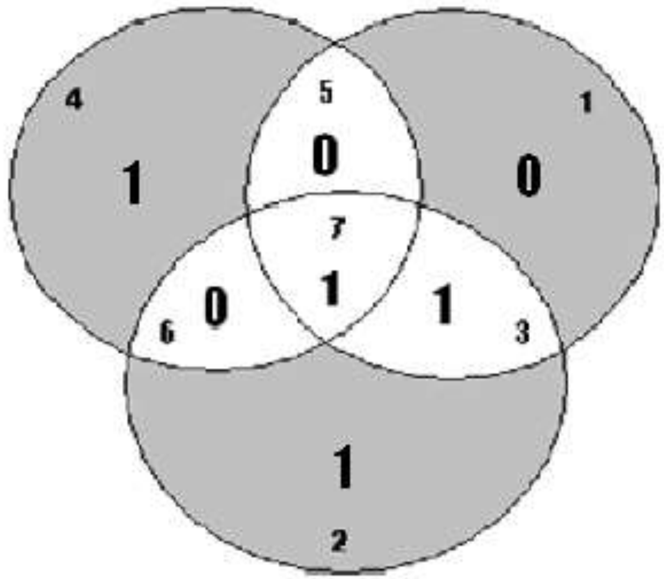
3

4

5

Step 21:

0 [1]	1 [2]	1 [3]	1 [4]	0 [5]	0 [6]	1 [7]



Decoder

Instruction for the animator	Text to be displayed in the working area (DT)
• The bits in the circle should appear one by one such that they are moving from rectangular box and placing in the respective circles. (in the sequential order)	• Downloading the bits into the Venn diagram

1

2

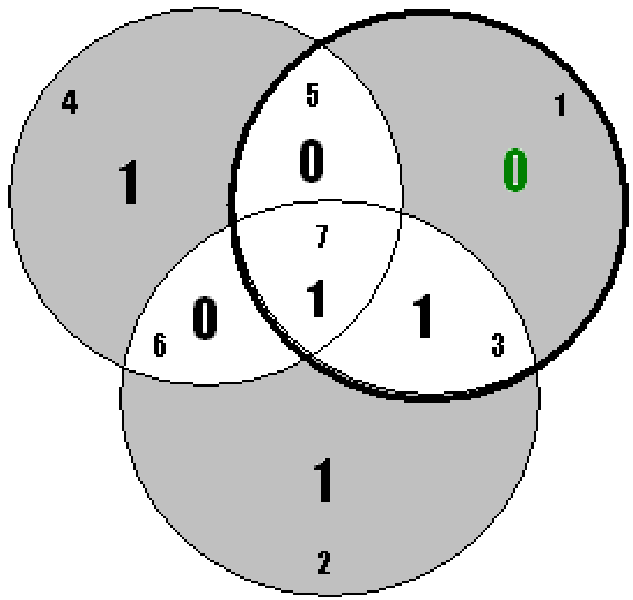
3

4

5

Step 22:

1	2	3	4	5	6	7
0	1	1	1	0	0	1



Decoder

Instruction for the animator	Text to be displayed in the working area (DT)
• Dark the circle first as shown and then blink 0s and 1s in 5,7,3 • Then show the 0 in green	• Checking for parity bit 1

1

2

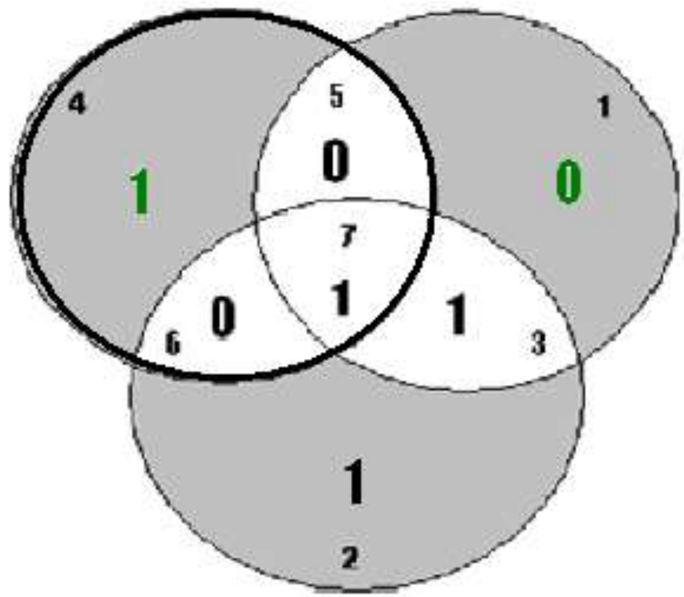
3

4

5

Step 22:

0 [1]	1 [2]	1 [3]	1 [4]	0 [5]	0 [6]	1 [7]

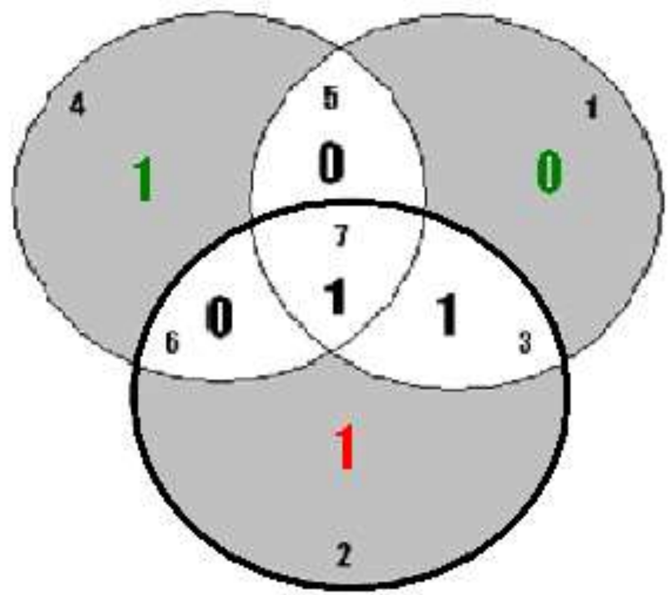


Decoder

Instruction for the animator	Text to be displayed in the working area (DT)
• Dark the circle first as shown and then blink 0s and 1s in 5,6,7 • Then show the 1 in green	• Checking for parity bit 4

Step 23:

0 [1]	1 [2]	1 [3]	1 [4]	0 [5]	0 [6]	1 [7]



Error Located

Instruction for the animator	Text to be displayed in the working area (DT)
• Dark the circle first as shown and then blink 0s and 1s in 3,6,7 • Then show the 1 in red • After red 1 show the statement " error located" • After showing 1 in red show the second sentence	• Checking for parity bit 2 • Modulo – 2 addition fails here • So there is an error

1

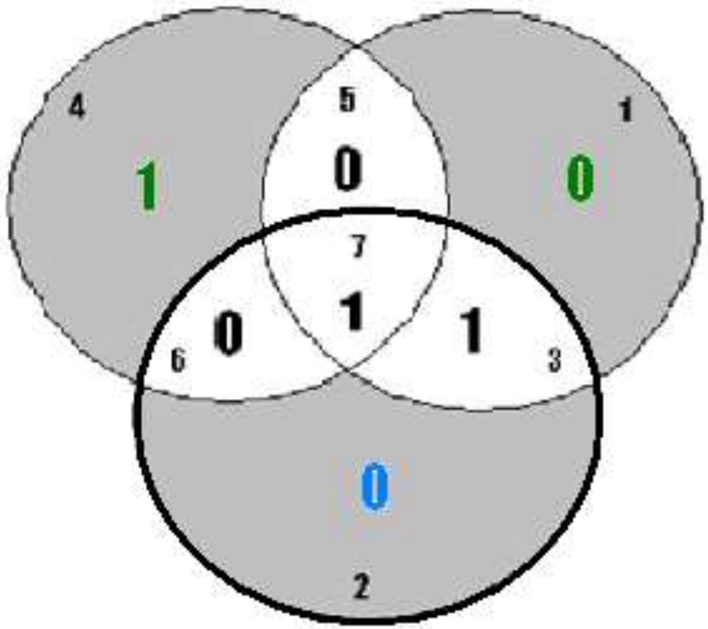
2

3

4

5

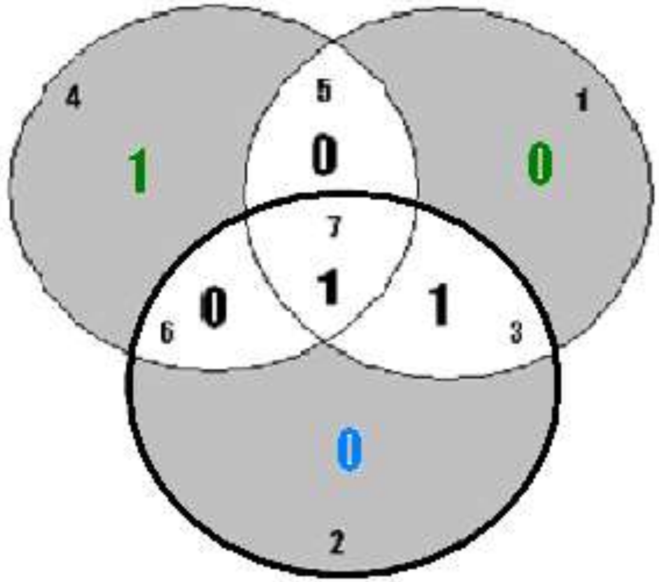
Step 24:



Error corrected

Instruction for the animator	Text to be displayed in the working area (DT)
• Then show the red 1 converted to blue 0 • After blue 0 is shown, the statement " error corrected" must be shown	• So in this manner error can be detected and corrected

Step 25:



1	2	3	4	5	6	7
0	0	1	1	0	0	1

Decoder output : 1 0 0 1

Instruction for the animator	Text to be displayed in the working area (DT)
• After the blue 1 is shown, show the rectangular box with red 1 in position 2 • Then show the red 1 converted to blue 0 • Then show the decoder output	• Decoding is completed

Enter the information bits :

3	5	6	7

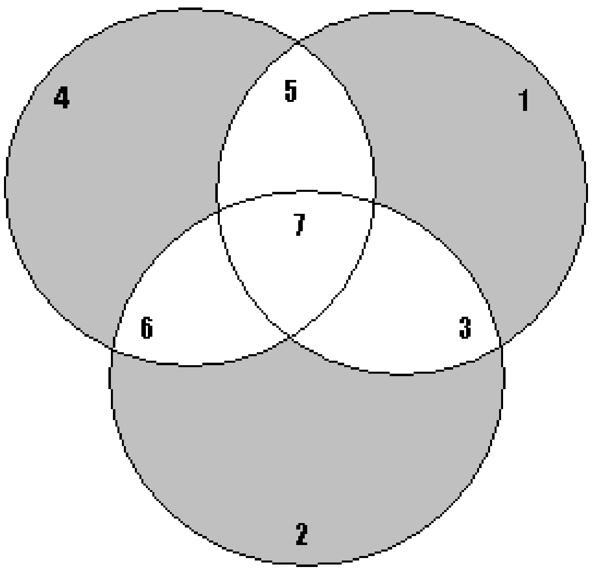
Interactivity:

Try it yourself

- Provide a box to enter the information bits

Calculating the parity bits:

1	2	3	4	5	6	7



Encoder

Interactivity:

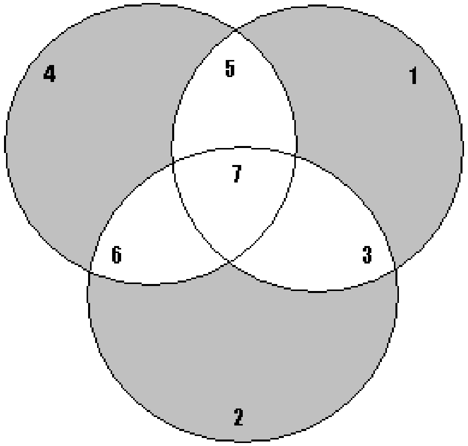
Try it yourself

- Provide a box to enter the information bits

Enter the errors:

1	2	3	4	5	6	7

Received codeword is:



Decoder

Decode output is:

Interactivity:

Try it yourself

- Provide a box to enter the information bits

Questionnaire

1 2 3 4 5

1. Hamming code can detect errors in information bits but not in parity bits

Answers: a) True b) False

2. What is the Syndrome if a single bit error happens in 6th location of a length 7 hamming codeword

Answers: a) 110 b) 101 c) 011 d)111

3. Hamming code can

i) detect any 2 –bit errors

ii) correct any 2 – bit errors

Which of the following is correct?

Answers:

a) i and ii are ture b) i is true but ii is false

c) i is false but ii is true d) i and ii are false

Links for further reading

Reference websites:

http://en.wikipedia.org/wiki/Hamming_code

http://www.ee.caltech.edu/EE/Faculty/rjm/SAMPLE_20040708.html

Books:

Error Control Coding – Shu Lin and Daniel J. Costello, Jr., second edition, Pearson

Research papers: