Binary Functions

Zero, Reset , Inhibit, Off

	0	1
0	0	0
1	0	0

One, Set, Assert, On

	0	1
0	1	1
1	1	1

And, Conjunction (XY)

	0	1
0	0	0
1	0	1

Nand ($\overline{XY} = \overline{X} + \overline{Y}$)

	0	1
0	1	1
1	1	0

$X \overline{Y}$ (set difference X – Y)

	0	1
0	0	0
1	1	0

$X + \overline{Y}$ (Y Þ X)

	0	1
0	1	1
1	0	1

$\overline{X}$

	0	1
0	0	0
1	1	1

X

	0	1
0	1	1
1	0	0

$\overline{X}\,Y$ (set difference $Y - X$)

	0	1
0	0	1
1	0	0

Implies $X \Rightarrow Y$ ($Y + \overline{X}$)

	0	1
0	1	0
1	1	1

Y

	0	1
0	0	1
1	0	1

$\overline{Y}$

	0	1
0	1	0
1	1	0

Exclusive or : XOR ($X \oplus Y$, $X \neq Y$)

	0	1
0	0	1
1	1	0

Equiv ($X \equiv Y$), XNOR

	0	1
0	1	0
1	0	1

(Inclusive) Or, disjunction (X+Y)

	0	1
0	0	1
1	1	1

Nor($\overline{X+Y} = \overline{X} + \overline{Y}$)

	0	1
0	1	0
1	0	0

Detection

Magic Cards

1	3	5	7	9	11	13	15
17	19	21	23	25	27	29	31
33	35	37	39	41	43	45	47
49	51	53	55	57	59	61	63

2	3	6	7	10	11	14	15
18	19	22	23	26	27	30	31
34	35	38	39	42	43	46	47
50	51	54	55	58	59	62	63

4	5	6	7	12	13	14	15
20	21	22	23	28	29	30	31
36	37	38	39	44	45	46	47
52	53	54	55	60	61	62	63

8	9	10	11	12	13	14	15
24	25	26	27	28	29	30	31
40	41	42	43	44	45	46	47
56	57	58	59	60	61	62	63

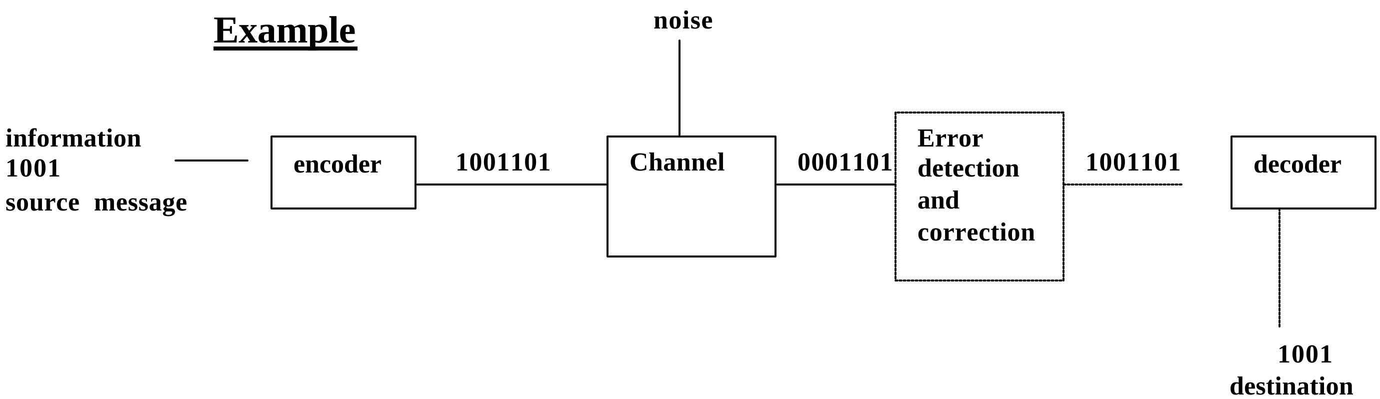
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63

32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63

Error detection and correction

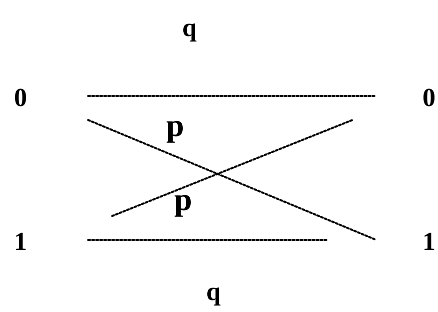
In a noisy channel errors may occur during the transmission of data from information source to destination, so we need a method to detect there errors and then correct them.

<u>Example</u>



Hamming Method

- **It was the first complete error-detecting and error-correcting procedure.**
- **It represents one of the simplest and most common method for the transmission of information (in the presence of noise)**
- **It assumes that the source transmits binary messages (i.e. The information source alphabet is { 0, 1 })**
- **It assumes that the channel is a binary symmetric channel**



- **It uses the parity checker method to detect an error**

Hamming

- ❖ Error-detection
- ❖ Parity checker

Hamming's single-error detecting code can be described as :

- ❖ **The first n-1 digits of the message are information digits**
- ❖ **In the n^{th} place we put either 0 or 1 so that the entire message has an even number of 1's.**
- ❖ **This is called an even parity check**

Example 1 **: Consider the message has an even number of 1's**

$$\begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \end{pmatrix}$$

with even parity check this message becomes **P**

$$\begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}$$

Example 2 : **given a 7-bit ASCII character $c_1, \ldots, c_7$ a parity check bit c_8 can be calculated as**

$$c_8 = (c_1 + c_2 + c_3 + c_4 + c_5 + c_6 + c_7) \bmod 2$$

$$= c_1 \text{ Å } c_2 \text{ Å } c_3 \text{ Å } c_4 \text{ Å } c_5 \text{ Å } c_6 \text{ Å } c_7$$

- ❖ **The parity check bits are calculated as the data is being written or transmitted, and then again as it is read or received, where if incorrect, some error-handling procedure is invoked.**

Hamming Error- Correction

- **It generalizes the idea of parity check, using multiple check bits to determine the position of an error (Then it can be corrected).**

- **A Hamming (7, 4) code means that there are**
  total 7 bits $\Rightarrow$ 4 for data + 3 for parity check

- **Let the total 7 bits labeled $c_1, c_2, c_3, c_4, c_5, c_6, c_7$**
  and c_1, c_2 and c_4 are the check bits and c_3, c_5, c_6, c_7 are the date bits

- **This method can indicate an error in any of the 7 locations hence we can correct it.**

- **The parity check bits c_1, c_2 and c_4 can be calculated as :**

$$c_1 = c_3 \oplus c_5 \oplus c_7$$
$$c_2 = c_3 \oplus c_6 \oplus c_7$$
$$c_4 = c_5 \oplus c_6 \oplus c_7$$

Example

For the message data

c_3	0 1 1
c_5	1 0 1
c_6	1 1 0
c_7	1 1 1

The Hamming (7, 4) code is

c_1	0 0 1
c_2	0 1 0
c_4	1 0 0